

# ROBOTICS AGE<sup>TM</sup>

The Journal of Intelligent Machines<sup>TM</sup> • NOV/DEC 1981 • \$3.00 USA/\$3.50 CANADA

**TEACH  
YOUR ROBOT  
TO SPEAK**



# Small, Dark and Handsome

## **RHINO** XR-1

HIGH-TECH ROBOT ■ FOR EDUCATION, RESEARCH & INDUSTRY

The Rhino® XR-1 high tech robot, though small in size, is built with the same operating technology as large industrial robots. And it features an RS232C interface (using 3 wires) making it compatible with the programming capabilities of any computer from the small PET®, Apple II®, or TRS-80® to the big IBM's. The Rhino® XR-1 is available for immediate delivery. Write for more information on your company letterhead to the address below.

#### Features:

- Digital Choppers on All Axes
- Powerful DC Servo Gear Motors
- Controller Card with Intel 8748 CPU
- Eight Motor Control Capability (Rhino® XR-1 has 6 motors)
- Operators Manual and Tool Kit
- Rigid ¼" Aluminum Space Frame Construction
- "Built-In" Functional Self-Test Program

#### Options:

- Custom Hands
- Computer Controlled Carousel, Conveyor, or Lateral Slide Base

#### Specs:

- Reach: 22.5" Maximum, 32" High
- Grasp: 16 oz. — open 2¼"
- Lift: 16 oz. at extension

#### Price:

\$2,400.00 (F.O.B. Champaign, IL.,  
Delivered via UPS in Continental USA)  
Illinois residents add 5% sales tax

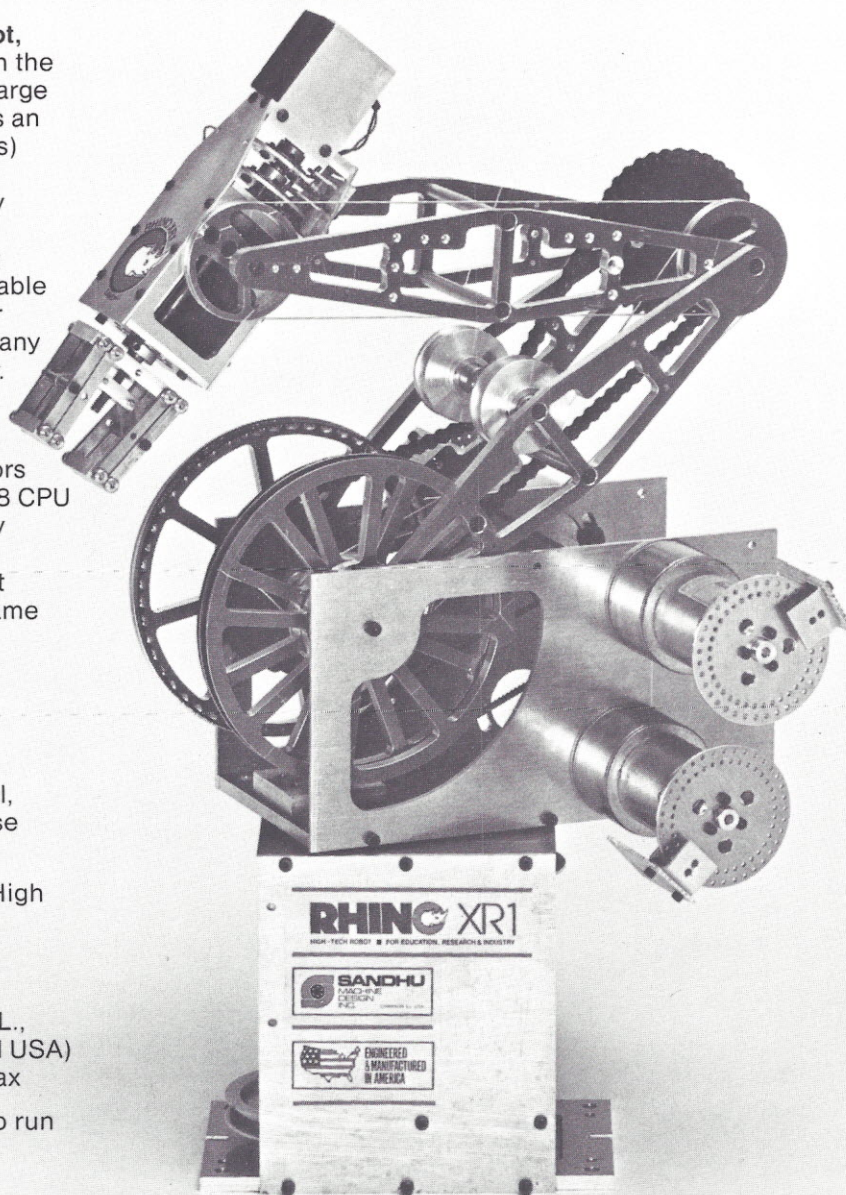
Includes everything you need to run  
your Rhino® XR-1 (Except your  
computer)

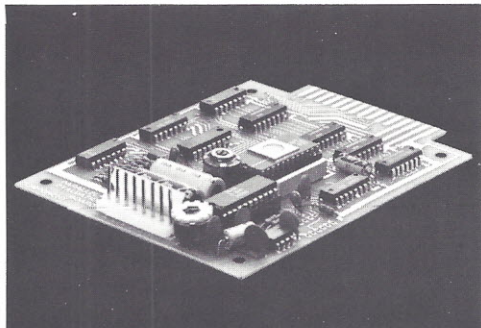
"Hands-On Introduction to ROBOTICS  
— The Manual for the XR-1" — \$35.00

We reserve the right to alter price  
and specifications without notice.

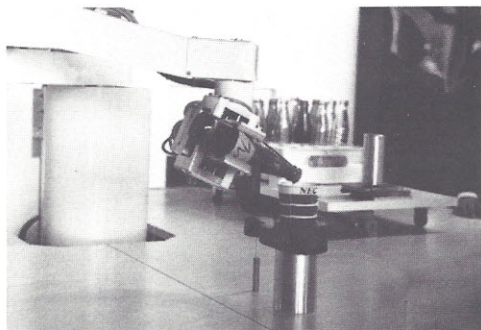


Sandhu Machine Design, Inc. Sales Dept.  
308 S. State, Champaign, IL. 61820  
(217) 352-8485

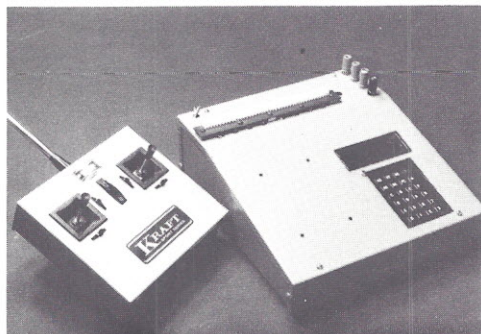




Page 4



Page 30



Page 37

## Articles

### 4 Teach Your Robot to Speak

by Tim Gargagliano & Kathryn Fons

With a phoneme synthesizer—and the techniques shown in this article—you can give your robot an unlimited vocabulary.

### 12 Fast Trig Functions for Robot Control

by Carl F. Ruoff

When you move your robot's arm to a particular point, your control program has to evaluate trig functions. Here's how to do it quickly.

### 22 An Interview with George Devol

by Jerry W. Saveriano

The inventor of industrial robots talks about the technical and social forces shaping this protean industry.

### 30 The Great Japanese Robot Show

by Joel M. Wilf

Robotics Age looks behind the scenes at the 1981 International Industrial Robot Exhibition.

### 37 TIMEL: A Homebuilt Robot, Part II

by John Blankenship

Under TIMEL's rough exterior, beats a heart of silicon. TIMEL's creator explains its control electronics.

## Departments

46 Eye to Industry

47 New Products

51 Media Sensors

53 Organizations

Cover photography by John R. Jones. Robot props courtesy of Gene Beley, Android Amusement Corp.

**Publisher**  
Philip C. Flora

**Managing Editor**  
Joel M. Wilf

**Art Director**  
Robert Tinney

**Technical Art**  
Heather Marr  
James Marr

**Advertising**  
Robotics Publishing Corp.  
10049 Commerce Ave.  
Tujunga, Calif. 91042  
(213) 352-7937

**Editorial Director**  
Alan M. Thompson

**Industrial Editor**  
Jerry W. Saveriano

**Production Editor**  
Cindee Huff  
**Production Assistant**  
Adrienne Wills-DeVine

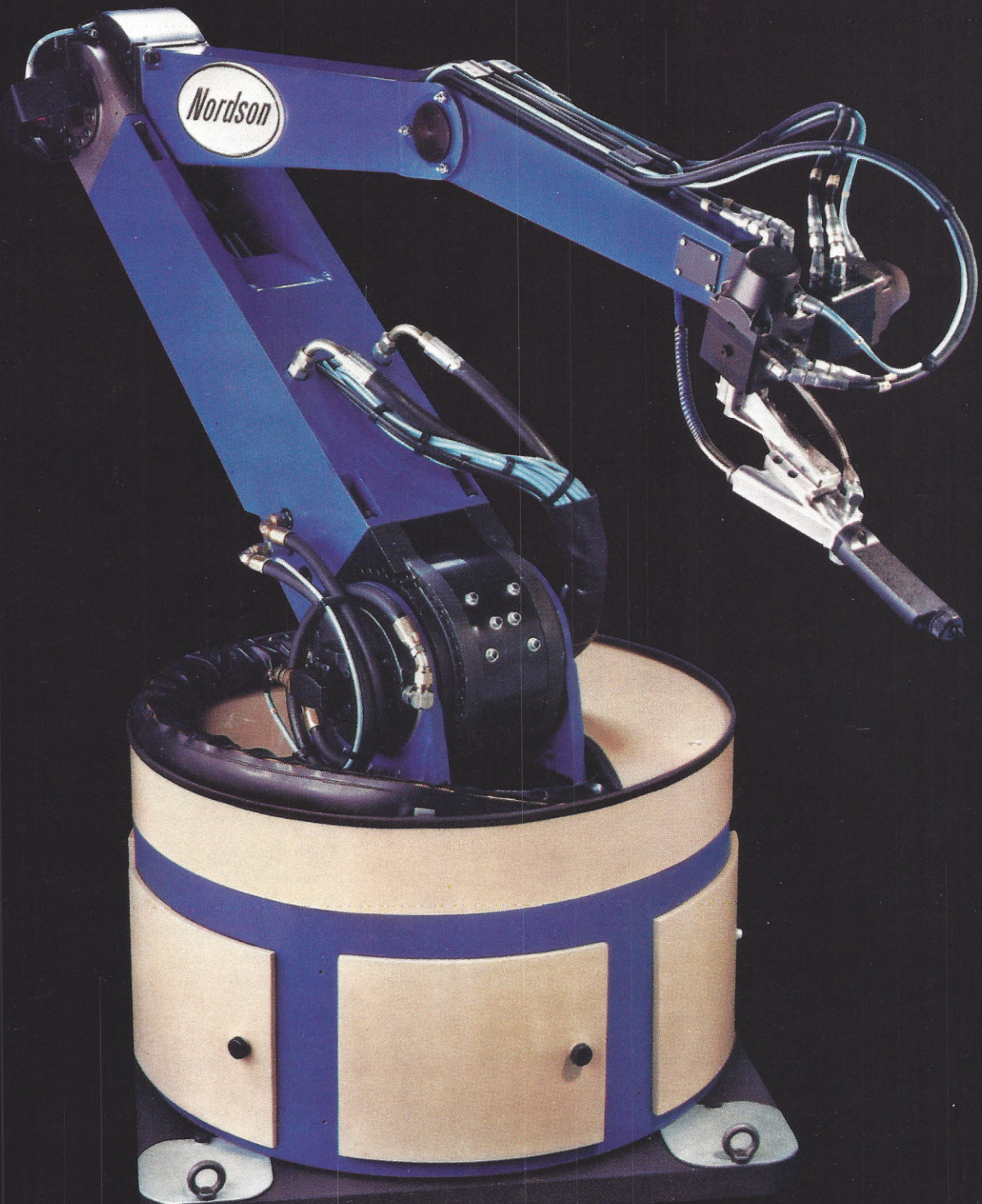
**Graphics**  
Steven Fagerquist

ROBOTICS AGE—USPS 523850 (ISSN 0197-1905) is published six times a year, in January, March, May, July, September, and November, by Robotics Publishing Corporation, 10049 Commerce Ave., Tujunga, California 91042. Subscriptions are \$15.00 for one year in the USA. Second-class Postage paid at Tujunga, CA 91042. POSTMASTER: Send address changes to P. O. Box 423, Tujunga, CA 91042.

Subscriptions from Canada from Mexico are \$17 for one year. Other foreign countries \$19 for one year surface delivery. Single copy price is \$3.00 in the U.S. and \$3.50 in Canada and Mexico, \$4 in Europe, and \$4.50 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a US bank. Address all correspondence to: Robotics Age, P. O. Box 423, Tujunga, CA 91042, phone 213/352-7937. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of ROBOTICS AGE. Each separate contribution to this issue, and the issue as a collective work

©1981 by Robotics Publishing Corporation. All rights reserved.

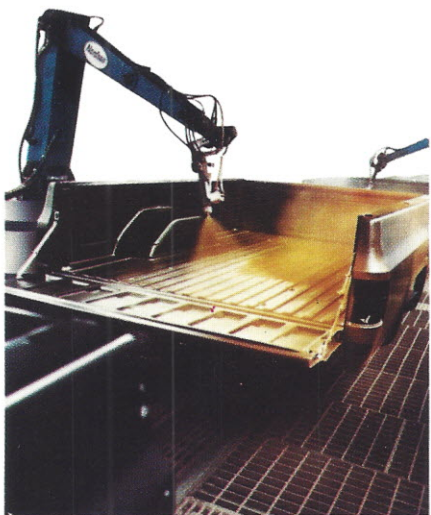
# More and more coats tailored by



# products wear Nordson® robots.

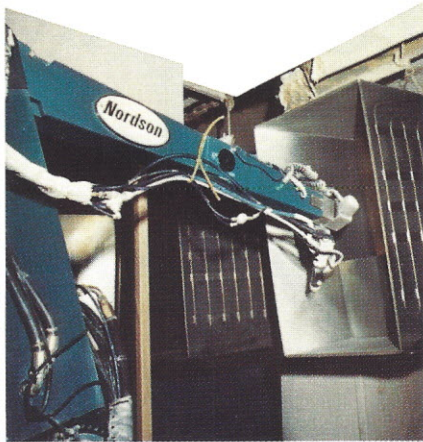
Nordson coating robots are dressing up an increasingly broad range of products—faster, safer, better and more economically.

Typical are pickup truck boxes rolling off the assembly lines of a major Ontario, Canada, automotive manufacturer. Two of our robots coat the interiors of the boxes with



better-than-ever finish quality... better-than-ever productivity... measurable savings in labor and materials.

At Franklin Manufacturing Company of St. Cloud, MN, one of the White Consolidated Industries, special clothing and equipment was needed to protect the health and safety of 15 painters spraying the inside of freezer liners because overspray bounced back out of the liners and onto the painters. Today, four Nordson robots do the same job, with



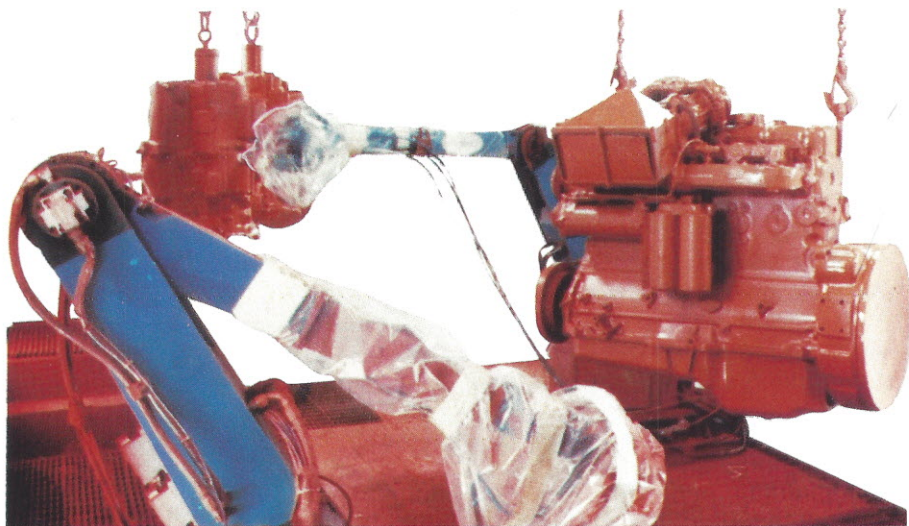
fewer rejects...no hazard to health and safety.

And at the Mack Truck manufacturing plant in Hagerstown, MD, two Nordson

paint...and without the need for manual touchup.

The burgeoning popularity of Nordson coating robots is due, in part, to the package of values that goes with them. A package that includes our worldwide network of staff engineers and technical service people. That includes tailoring a system to your exact needs, which alone could save you thousands of dollars a year. And that includes our in-depth factory training program, where you learn to work with robots.

We'd like you to visit us and see Nordson robot technology in action. To arrange your



robots working in tandem apply a protective coating of paint to truck engines. They provide the same complete coverage as with the previous method of application but accomplish it using 40% less

visit, or to get more information, call toll-free 1-800-321-1414 (in Ohio, call 1-800-362-9947), Ext. 751. Or write Nordson Corporation, Robotics Division, 555 Jackson Street, Amherst, OH 44001.



Robotics Division

# TEACH YOUR ROBOT TO SPEAK

by  
Tim Gargaliano  
and  
Kathryn Fons

Votrax Division Federal Screw Works  
Troy, Michigan

Machines have communicated with machines for a long time. But as robots enter our lives, how will they speak to us? Unlike machines, humans can't be connected to a standard RS232C or IEEE488 interface.

Traditionally, machines have given us information through display devices, such as the CRT. But displays have disadvantages. They require visual attention. And, in their present state of development, full-screen displays come in two flavors: fragile and cumbersome—or extremely expensive.

Why not have machines use the *spoken* word? We can listen to speech while doing almost anything else. In addition, current speech technology offers devices that are small, rugged, and inexpensive.

This article is about speech technology. It briefly covers the basics of synthesizing speech. It describes inexpensive hardware that can give an unlimited vocabulary to your robot. And it explains the programming techniques you need to teach your robot how to speak.

## Synthesizing Speech

There are two fundamentally different ways to synthesize speech. In the first method, we reconstruct a *time domain* signal from digital samples taken from a human source. As Figure 1 shows, a time domain signal records how the sound's loudness varies with time. The amount of data depends on the rate at which we sample the signal and the number of quantized levels used in the analog-to-digital conversion. Data compression techniques can reduce the overall bit rate. But time domain synthesis still needs between 16,000 and 64,000 bits/second.

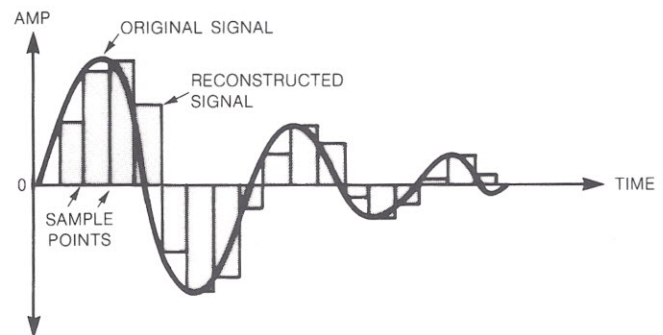


Figure 1. Reconstructing a signal in the time domain.

In the second approach, we synthesize the frequency content of speech as it varies with time. Frequency information for speech is contained in bands called *formants*, shown in Figure 2. During dynamic speech, these formants change in their center frequency and bandwidth. Since *frequency domain* information changes much more slowly than time domain information, this technique needs much less data. Typically, a formant synthesizer has a data rate of between 1000 and 2000 bits/second.

A formant synthesizer, shown in Figure 3, consists of a filter network, plus pitch and noise sources that excite the filters. In this system, filter parameters and excitation amplitudes are under external control.

You could control the formant synthesizer by individually setting each of its parameters. But suppose these parameters naturally occurred in certain combinations. Then, we could encode these combinations. We could feed the code to a parameter generator/interpolator which, in turn, would supply a continuous flow of

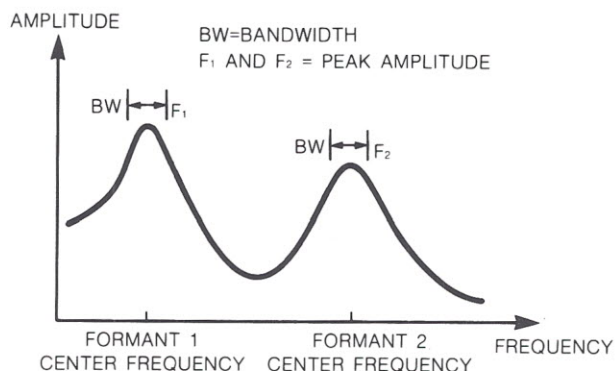


Figure 2. A formant plot shows the frequency content of speech at a given instant.

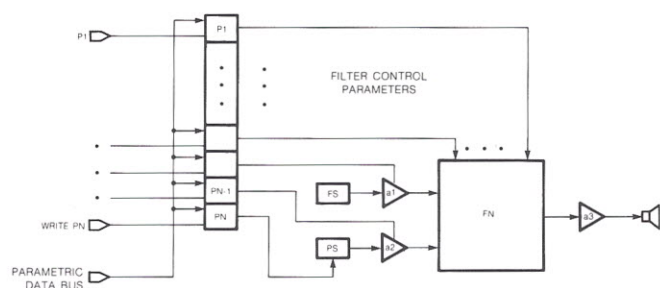


Figure 3. Block diagram of a speech formant synthesizer.

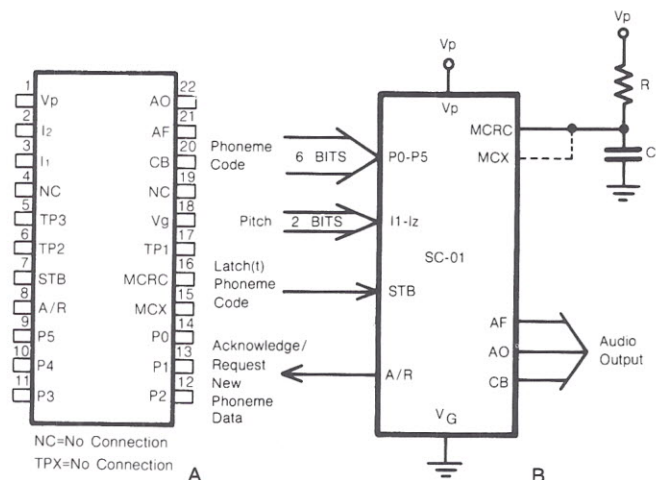


Figure 4. The SC01 speech synthesizer chip: (a) pin out, (b) functional diagram.

parametric data to the formant synthesizer.

Certain combinations of formant parameters do occur in human speech. These are *phonemes*, the basic speech sounds we use to create words. A *phoneme synthesizer* uses phoneme codes to control a formant synthesizer. The main advantage of this approach is its low data rate—typically under 100 bits/second. At this rate, a 2716 EPROM could store about 200 words built from phonemes.

### The SC01 Phoneme Synthesizer Chip

The Votrax SC01 is the first single chip phoneme synthesizer. The SC01 contains an analog model of the human vocal tract, using switched capacitors to emulate resistors for a small chip size. During continuous speech, the SC01 needs only about 70 data bits per second—though this figure varies with the master clock frequency. As shown in Figure 4, the chip comes in a 22 pin DIP. Since it uses CMOS logic, its power consumption is low enough for battery-powered systems.

As you can see in Figure 5, the block diagram of the SC01, the chip consists of a latch (for bits P0-P5), a phoneme controller, pitch and noise sources, a filter network (simulating the vocal tract), an audio preamplifier, handshake logic, and an onboard oscillator (master clock). The master clock determines all internal timing functions. Its frequency should be set to about 720 KHz by connecting an external resistor and capacitor to the MCRC (master clock resistor/capacitor) pin. As an alternative, you can supply an external master clock to the MCX pin. By varying this clock's frequency, you can create different voices and sound effects.

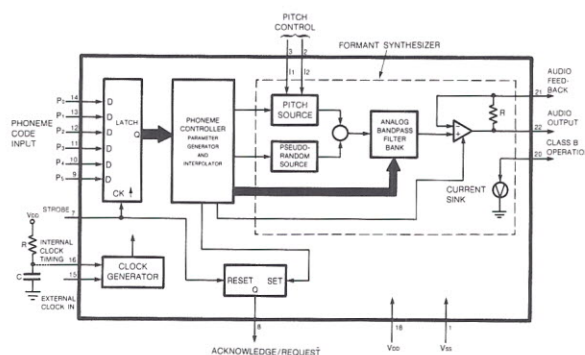
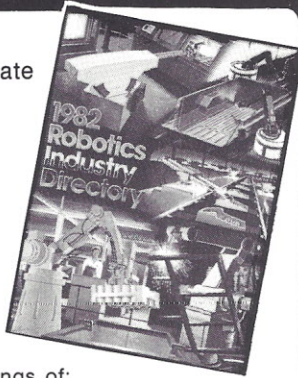


Figure 5. Internal block diagram of SC01.

# ROBOTS 1982

- Over 300 listings
- Now with monthly update

## The 1982 Robotics Industry Directory



contains comprehensive listings of:

**INDUSTRIAL ROBOTS**, worldwide, with prices, complete specifications and the name, address, and phone number of a knowledgeable representative. Specification matrices are included for quick side-by-side comparison of competitive models. **ROBOT COMPONENTS & PERIPHERALS**, including intelligent parts feeders, positioning mechanisms, vision systems, controllers, and more. **CAM SYSTEMS** that can utilize robots. **ROBOT DISTRIBUTORS**, including robot systems houses. **CONSULTANTS** with robotics expertise. **ROBOT RESEARCH INSTITUTES...PLUS MORE!**

\*\*\* ALL THE INFORMATION YOU NEED  
TO SUCCESSFULLY EVALUATE INDUSTRIAL ROBOTS  
IN ONE EASY-TO-USE, COMPREHENSIVE VOLUME \*\*\*

**Order Today—Only \$35 (\$60 with monthly update)**

(Canada \$37US, \$67US with update; other Foreign \$43US,  
\$83US with update—airmail delivery included)

Robotics Publishing, Dept. M, PO Box 423, Tujunga, California 91042

CIRCLE 26

The input latch on pins P0-P5 hold the phoneme code, which is fed to the phoneme controller. Here, the chip generates parameters to dynamically control the "vocal tract" and the pitch and noise sources. The output of the vocal tract is preamplified before it exits the audio output pins. Two additional pins—audio feedback (AF) and class B (CB)—provide for external amplification.

At the heart of the SC01, a phoneme synthesizer produces 64 speech sounds. In English, at least 45 phonemes are needed to produce every word in the language. The additional 19 sounds, called *allophones*, are variant pronunciations of certain phonemes. Table 1 lists the 64 sounds produced by the SC01, along with key words that contain each sound. By selecting and sequencing these phonemes—writing phonetic programs—you can teach the speech synthesizer to speak.

### How to Program Phoneme Sequences

As a phonetic programmer, you'll create a sequence of phonemes, which the synthesizer will output as speech. The better you program the sequence, the clearer, more accurate, and more human the synthesizer sounds.

How can you translate a word into sounds? You could begin by looking up the written word in the dictionary. But most written characters have more than one possible pronunciation. Therefore, we need an alphabet—different from the written system—that represents phonemes.

Each dictionary has such an alphabet, defined by its "pronunciation key." Also, linguists have created a standard International Phonetic Alphabet (IPA). Votrax uses the phoneme symbols given in Table 1. For this article, we'll stay with Votrax's phoneme symbols. Whenever we use phoneme symbol that might be confused with a written character, we'll enclose the phoneme in slash marks. For instance, /A/ means the vowel sound in the word "tame."

Now, with grammar rules and a dictionary in hand, you can begin phonetic programming. The programming procedure typically consists of the following eight steps:

1. Select a word to synthesize.
2. Pronounce the word aloud.
3. Count the number of sounds in the word.
4. Translate the word into a phoneme sequence. For each sound counted in step 3, find its appropriate phoneme symbol, using Table 1.
5. Enter the phoneme sequence into the synthesizer, and activate the speech output.
6. Listen to the synthesizer's output. Check how accurately it pronounces the word you programmed.
7. Adjust the phoneme sequence wherever the synthesizer pronounces inaccurately.
8. Repeat steps 6 and 7 until you have the speech output you want.

Step 7 is where you'll spend most of your programming time. This is because step 4 only gives the primary sound sequence of a word. In order to match human pronunciation—which injects rhythm and meaning into a word—you have to adjust the sequence to add stress.

In grammatical rules, stress is often shown as an accent placed on a portion of the word. When programming a phoneme sequence, you can add stress by increasing the duration, pitch, or amplitude of the vowel sound in the accented part of the word. Table 2 shows how the words "family," "sensory," and "visual" advance from an initial program, through trial sequences, to the final refined program. The major difference is stress.

Notice that we start by placing equal stress on all syllables in the initial program. To begin with, we use the longest lasting vowel sound that translates the vowel letter. During the trial sequencing, we substitute shorter lasting vowel sounds in the unaccented syllables.

The key to achieving a refined program is selecting the appropriate vowel durations for each syllable. This must be done by listening closely to the speech output. The durational relationship of one vowel to another must sound correct. In the word "family," for example, the first syllable lasts the longest, while the last syllable is of medium duration. The middle syllable has the shortest

duration. It is virtually absent in the best pronunciation of "family."

### Advanced Techniques

Sometimes you can select phonemes more accurately if you know how the human vocal tract produces them. The study of how humans produce sound is called *articulatory phonetics*. When you make a sound—say the vowel sound /AH/—notice how you do it. Notice that your mouth is wide open and that the sound resonates from the back. In terms of articulatory phonetics, this means that /AH/ is an *open back vowel*. Table 3 shows some articulatory features of vowel phonemes and allophones.

This information is helpful when you want to program *diphthongs*—two or more vowel sounds in sequence, spoken as a single unit. In a diphthong, the tongue moves rapidly from one position to another. By noting how you make each vowel sound in the diphthong and referring to charts like table 3, you can select the right phonetic representation.

Take the sound of the long "i" in the word "time," for example. When you say the long "i" diphthong, notice how your tongue starts in the back of your mouth, then travels quickly to the front as your lips close slightly. This means that the first sound in the diphthong is an open back vowel, and the second sound is a closed front vowel. Using table 3, we combine /AH/ and /E/ to create the sound.

This gives us a first approximation to the sound of "i" in "time." But after you listen to the synthesizer say the sequence, you'll realize that we need shorter lasting sounds in order to perceive the sequence as a single unit. This leads us to select the allophones /AH1/ and /E1/ to produce the sound.

Once again, you'll feed the combination to the synthesizer. When you hear the output, you'll notice that in a one syllable word, like "time," the "i" sound should be slightly longer in duration. Using table 3, we can select a vowel that falls between our original two sounds and insert it into the sequence. The /EH3/ qualifies, as does the /I3/. The former sound favors the more open production of /AH/, while the latter favors a more closed production, like /E/. Either selection could work. Choose the one that sounds best to you.

It's easier to program consonants than vowels. Still, there are cases that articulatory phonetics can help. Take, for example, *stop-plosive* sounds. These are the sounds produced when we block the vocal tract (with our lips, for instance), then suddenly release the sound. /B/, /D/, /P/, and /T/ are examples of stop-plosives.

TABLE 1

VOTRAX PHONEME TABLE AND PRONUNCIATION KEY			
CONSONANTS		VOWELS	
PHONEME SYMBOL	KEY WORD	PHONEME SYMBOL	KEY WORD
B	Bat - ruB	A	tAme - pAl1 - mAke
CH	CHeese - maTCH	A1, A2	
D	DaD - raID	AE	sAd - pAlId
DT	buTTer	AE1	
F	Fake - cuFF	AH	tOp - fAther
	PHone - lauGH	AH1, AH2	cAl1 - OfFice
G	Get - loG	AW	
H	Have - Help	AW1, AW2	
J	Jazz - JuDGe	AY	tAme - pAl1 - stAY
K	Car - KICK	E	bEEf - Even
L	Low - caLL	E1	
M	Mat - diM	EH	IEg - rEAdy - sAlId
N	No - soN	EH1, EH2,	
NG	riNG - driNk	EH3	
P	Pack - haPPy	ER	biRd - hEARd - ovER
R	Race - haRd	I	pl1 - swIm
S	Soup - lightS -	I1, I2, I3	
	City - aSk	IU	YOU - mUsic
SH	SHeeP - acTion	O	fOr - tOrn - bOlId
T	Tip - askeD	O1, O2	
TH	THing - maTH	OO	tOOk - pUt - cOUId
THV	THe - moTHer	OO1	
V	Van - paVe	U	mOve - schOOl - JUne
W	Wake - WHen -	U1	
	qUit	UH	cUp - ARound - sOn
Y	Yard - berrY	UH1, UH2	
Z	Zap - haZe -	UH3	randOm - statIOn
	panS - glasseS	Y1	You - mUsic
ZH	pleaSure - aZure		
PA1	(long pause)		
PA0	(short pause)		
STP	(no sound)		

TABLE 2

WORDS	INITIAL PROGRAM	TRIALS	REFINED PROGRAM
FAMILY	F-AE-M-I-L-E	F-AE-M-I3-L-E1 F-AE1-M-I3-L-E1 F-AE1-EH3-M-I3-L-E1	F-AE1-EH3-M-L-E1
SENSORY	S-EH-N-S-O-R-E	S-EH1-N-S-O2-R-E1 S-EH1-N-S-ER-E1	S-EH2-EH3-N-S-ER-E1
VISUAL	V-I-ZH-U-UH-L	V-I1-ZH-U1-UH3-L	V-I1-ZH-W-UH3-L

TABLE 3

VOWEL PLACE-OF-PRODUCTION CHART			
FRONT VOWELS	MEDIAL VOWELS	BACK VOWELS	MOUTH
E		U	CLOSED ↑ ↓ OPEN
E1		U1	
I		OO	
I1, I2, I3		OO1	
	UH	O	
A	UH1, UH2, UH3	O1, O2	
A1, A2		AW	
EH		AW1, AW2	
EH1, EH2, EH3		AH	
AE		AH1, AH2	
AE1			OPEN

Stop-plosives are difficult to perceive, even when spoken by a human speaker. When synthesizing these sounds, it is helpful to double the consonant sound, or combine it with its *voiced* (uses the vocal chords) or *voiceless* (doesn't use the vocal chords) counterpart.

# Introducing:

# SmartWare<sup>TM</sup>

(LISP) 1.7 & 2.0

Introducing  
LISP. The language  
that can think for itself.

With SmartWare, your micro-computer possesses intelligence. Solves problems like never before. Actually can learn from its mistakes. And educates itself in much the same way your brain operates. It's a new concept in the way information is handled in microcomputers.

We appropriately called our version of LISP, SmartWare. Because it's as limitless as the human imagination, mind, memory.

An advanced, high-level language, LISP was first developed for use in artificial intelligence on large computers. And now, LISP is available from Datasoft for use on the Apple II and Atari Computers.

Store multiple programs in memory. It can take it. Program other languages into LISP. It's no problem. Because LISP offers greater capacity and power. And, it's a faster, more streamlined language. For example, just 2 lines in LISP could equal hours of BASIC programming.

The fact: Relational data base capabilities. User and program definability. Pattern-directed

invocation language. Uses syntax and data structures upon which esoteric applications may be implemented. Remembers data along with "relationships affecting it." Offers REAL power to micros.

At MIT, they say LISP is the language of the future.

At datasoft, we say, why wait. Ask your local computer dealer for SmartWare. Right now! And turn your computer on to thinking.

**SmartWare by**  
**Datasoft Inc.<sup>TM</sup>**  
COMPUTER SOFTWARE

19519 Business Center Dr., Northridge, CA 91324  
(213)701-5161

The programs for "eighteen" and "ninety" are examples of this. "Eighteen" has the sequence A1-AY-Y1-T-T-E1-N. This stop-plosive /T/ is doubled. "Ninety" is programmed as N-AH1-I3-Y-N-D-Y. In this case, the stop-plosive /T/ is replaced by its unvoiced counterpart, /D/. Table 4 shows how the vocal tract produces these and other consonants.

Given the subtle problems just discussed, you'll find it helpful to use established phoneme sequences as patterns for creating new words. Table 5 lists some words that apply to robotics, along with their phonetic programs.

TABLE 4

CONSONANT MANNER and PLACE -OF-PRODUCTION CHART		
MANNER-OF-PRODUCTION		PLACE-OF PRODUCTION
VOICED	VOICELESS	
B,D,G	P,T,K,DT	STOP PLOSIVES
Z,ZH,V,THV	S,SH,F,TH,H	FRICATIVES
J	CH	AFFRICATES
M,N,NG		NASALS
R,L,W,Y		SEMIVOWELS/GLIDES

TABLE 5

SAMPLE WORD LIST	
PHONETIC PROGRAM	
arm	AH1-UH3-R-M
backward	B-AE1-EH3-K-W-ER-D
come	K-UH1-UH3-M
computer	K-UH1-M-P-Y1-IU-U1-T-ER
down	D-AH1-UH3-U1-N
emergency	I2-M-R-R-D-J-I2-N-S-Y
find	F-AH1-EH3-Y-N-D
forward	F-O2-O2-R-W-ER-D
go	G-OO1-O1-U1
hello	H-EH1-UH3-L-UH3-O1-U1
help	H-EH1-UH3-L-P
human	H-Y1-IU-U1-M-EH3-N
left	L-EH1-EH3-F-T
leg	L-EH1-I3-G
me	M-E1-Y
no	N-OO1-O2-U1
noise	N-O1-UH3-I3-AY-Z
off	AW2-AW2-AW2-F
on	AH1-UH3-N
please	P-L-E1-Y-Z
right	R-UH3-AH2-Y-T
robot	R-O2-O2-B-AH1-UH3-T
stop	S-T-AH1-UH3-P
up	UH1-UH2-P
yes	Y1-EH3-EH1-S

There are no steadfast rules for phonetic programming, only pronunciation guidelines. You're free to experiment with different phoneme combinations to create a special effect. For example, you may want to hear a certain dialect of English or a word from a foreign language. In either case, you need only follow the eight-step procedure that we outlined earlier.

### Special Effects

You can produce different voices and sound effects simply by varying the master clock frequency to the SC01. With slow minor variations, you can create various voices. Fast drastic changes in frequency produce sound effects.

In either case, you might want to control these changes with software from a computer. There are several ways to do this. One is to vary the frequency of the SC01 master clock oscillator by switching in different resistor/capacitor combinations. A circuit that does this is shown in figure 6. An 8 bit binary weighted resistor ladder can be selectively connected to a fixed capacitor and the SC01. A CD4016 analog switch is used to select each resistor in the ladder into or out of the circuit. As more resistors are switched into the circuit, their effective resistance becomes lower and the master clock frequency becomes higher.

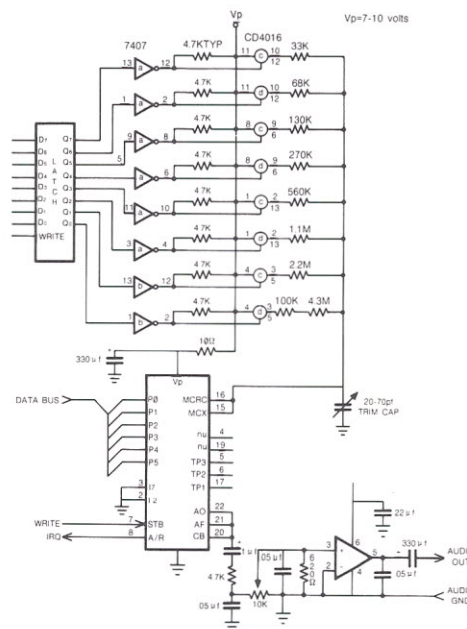


Figure 6. An SC01 application circuit. The switched resistor network can vary the clock frequency to generate sound effects.

```

EXAMPLE RAMP SOUND EFFECT ROUTINE

; Generate a ramp number sequence
;
; B = repeats   C = increment size
; D = start     E = stop
; HL = timer counts between increment

REPEAT: LD    A,D      ; A = start number
OUTPUT: OUT   (LATCH),A ; output current number
        PUSH  HL       ; save timer value
        CALL  TIMER    ; wait for selected interval
        POP   HL       ; recover timer value
        ADD   C         ; add increment
        CP    E         ; at end?
        JR    NC, OUTPUT ; brif not
        DJNZ  REPEAT   ; repeat ramp
        RET

LATCH: EQU 10 ; output port location
TIMER: EQU 1000H ; delay subroutine location

```

Figure 7. Subroutine to produce a ramp sound effect. (Timer subroutine and output latch locations may vary.) The ramp modulates the phonemes output by the SC01.

```

1 ; use the data setting directives found on
2 ; an assembler to pack the phoneme lexicon
3
4 ARM:      DC  AH1, UH3, R, M+FLAG
5 BACKWARD: DC  B, AE1, EH3, K, W, ER, D+FLAG
6 COME:     DC  K, UH1, UH3, M+FLAG
7 COMPUTER  DC  K, UH1, M, P, Y1, IU, U1, T, ER+FLAG
8 DOWN:     DC  D, AH1, UH3, U1, N+FLAG
9 EMERGENCY: DC I2, M, R, R, D, J, I2, N, S, Y+FLAG
10 FIND:    DC  F, AH1, EH3, Y, N, D+FLAG
11 FORWARD: DC  F, O2, O2, R, W, ER, D+FLAG
12 GO:      DC  G, OO1, O1, U1+FLAG
13 HELLO:   DC  H, EH1, UH3, L, UH3, O1, U1+FLAG
14 HELP:    DC  H, EH1, UH3, L, P+FLAG
15 HUMAN:   DC  H, Y1 IU, U1, M, EH3, N+FLAG
16 LEFT:    DC  L, EH1, EH3, F, T+FLAG
17 LEG:     DC  L, EH1, I3, G+FLAG
18 ME:      DC  M, E1, Y+FLAG
19 NO:      DC  N, OO1, O2, U1+FLAG
20 NOISE:   DC  N, O1, UH3, I3, AY, Z+FLAG
21 OFF:     DC  AW2, AW2, AW2, F+FLAG
22 ON:      DC  AH1, UH3, N+FLAG
23 PLEASE:  DC  P, L, E1, Y Z+FLAG
24 RIGHT:   DC  R, UH3, AH2, Y, T+FLAG
25 ROBOT:   DC  R, O2, O2, B, AH1, UH3, T+FLAG
26 STOP:    DC  S, T, AH1, UH3, P+FLAG
27 UP:      DC  UH1, UH2, P+FLAG
28 YES:     DC  Y1, EH3, EH1, S+FLAG
29 FLAG:    EQU 1000000B ; last phoneme flag bit
30

```

Figure 8. Creating a lexicon. This code assumes that the EQUates for the SC01 phoneme symbols have already been done.

Figure 7 shows a software routine that generates a "ramp" sound effect using this resistor ladder. By selecting different phonemes—holding the phoneme steady during the whole effect—you can get a panoply of different sounds.

As you generate a vocabulary or accumulate phoneme programs, you may want to create a lexicon (word table) to be stored on disk or in ROM. Figure 8 gives a Z-80 assembly language program that packs words into a lexicon. Figures 9 and 10 give SC01 calling and driving routines. (The driving routine is intended to be interrupt driven.)

```

MSG1: LD  DE,PLEASE ; DE points to please
      CALL SPEAK    ; speak
      LD  DE,GO     ; DE points to go
      CALL SPEAK    ; speak
      LD  DE,LEFT   ; DE points to left
      CALL SPEAK    ; speak
      RET

SPEAK: LD  A,(DONE) ; speech busy?
      CP  0         ;
      JR  Z, SPEAK  ; brif yes
      LD  (NXTPH), DE ; save pointer
      LD  A,0       ;
      LD  (DONE),A  ; set busy flag
      RET

```

Figure 9. An example of a speech calling procedure. A call to SPEAK sets the pointer NEXTPH to the next phoneme string and sets the flag DONE to false (0). On the next interrupt from the SC01, phoneme output will begin.

```


SRV-SC01: LD  A, (DONE) ; speech pending?
          CP  0         ;
          JR  NZ, .SRVC2 ; brif not
          LD  DE, (NXTPH) ; load pointer
          LD  A, (DE)    ; get a phoneme
          INC DE
          LD  (NXTPH), DE
          BIT  7, A      ; test for flag bit
          JR  Z, .SRVC1 ; brif not
          LD  (DONE), A  ; set flag
          .SRVC1 OUT (SC01), A ; output phoneme
          RETI
          .SRVC2 LD  A, PA1 ; A= pause code
          OUT (SC01), A ; output pause
          RETI
          DONE DC  0

```

Figure 10. SC01 interrupt routine. It outputs the next phoneme (pointed to by NEXTPH) and increments the pointer. It sets DONE to true if it's the last phoneme.

## Conclusion

Robots that talk are now a reality. Technical advances in the last five years have shrunk speech synthesizers to the size of an LSI chip. Complete lines of speech-based products have been put on the market. (See the sidebar on "Speech Products.")

We are entering an era of talking machines. Someday soon, our cars will tell us when they need gas, oil, or repairs. Our ovens will tell us how to make our favorite recipes. Our hobby kits will tell us how to build them. Our games will tell us how to play them. For the average consumer, talking machines may still be a few years in the future. But for those with a computer and a speech synthesizer, that future is now. 

## Speech Products

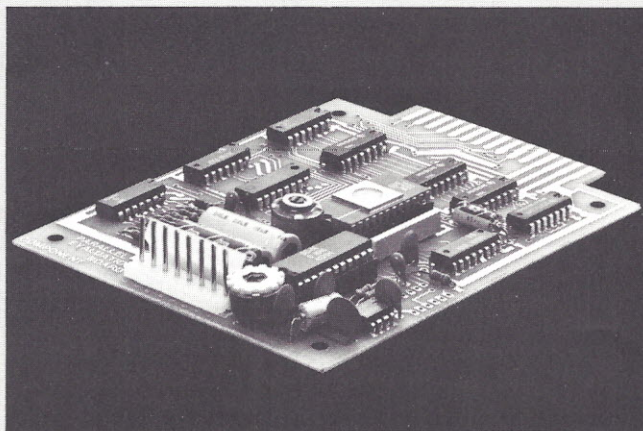
Phonetic programming—though easy once you learn the routine—remains a very time consuming task. There are several speech-based products designed to make life easier for you. The examples we give here are all currently offered by Votrax.

The Votrax Speech PAC board, shown in figure (a), is built around the SC01 speech chip. In addition, it contains a parallel interface, a 1 watt audio amplifier, and a prestored vocabulary of 250 words. If you need to modify any of these words, the PAC board allows you to override any of the stored phonemes.

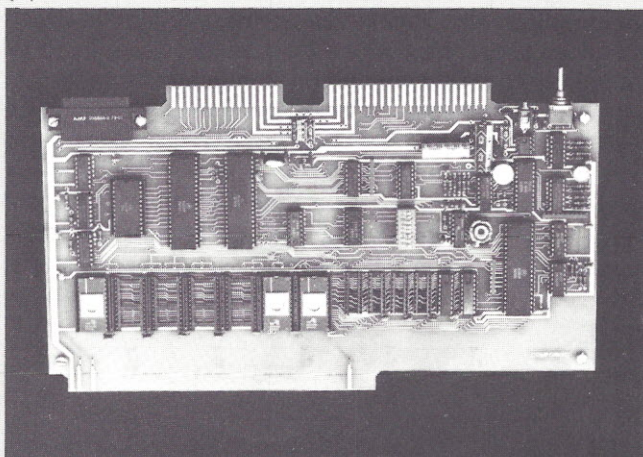
For more advanced applications, there is the Votrax VSM/1 (Versatile Speech Module), shown in figure (b). This board contains an SC01 speech chip, an M6800 processor with support circuitry, an RS232C interface, a parallel interface, a 1 watt audio amplifier, sound effect circuitry, 1300 words of prestored vocabulary, 1K byte of RAM, and a terminal operating system. The VSM/1 can serve as a board for experimenting with speech and sound effects. Or it can work as a stand alone controller (as in industrial applications).

Those who want to avoid phoneme programming can use a text-to-speech translator. The Votrax Type 'N Talk, shown in figure (c), automatically translates a string of ASCII characters into phoneme sequences. Using its own text-to-speech algorithm, it connects to any RS232C compatible source of ASCII codes. You can send the output phoneme strings to either the onboard SC01 speech synthesizer or back to the host computer. (This device is described in "New Products," *Robotics Age*, May/June 1981.)

(a)



(b)



(c)



# Fast Trig Functions for Robot Control

by  
Carl F. Ruoff  
Robotics Research Program  
Jet Propulsion Laboratory  
California Institute of Technology  
Pasadena, California

How do you tell your robot where to go? If you're controlling a robot manipulator directly—with a joystick, for example—you can watch the result of your commands and use your own visual feedback to get the arm on target. If the robot is recording the target points you teach it as a programmed movement sequence, it must have a way to represent each target so that it can return to it later. If the robot has position feedback sensors in the arm, (encoders, pots, etc.) it can simply read the position of each joint or linkage and store these values. (See Figure 1). Later it can move the arm to the identical configuration by matching the recorded values. As long as neither the target point nor the robot has moved, the arm will be at the target.

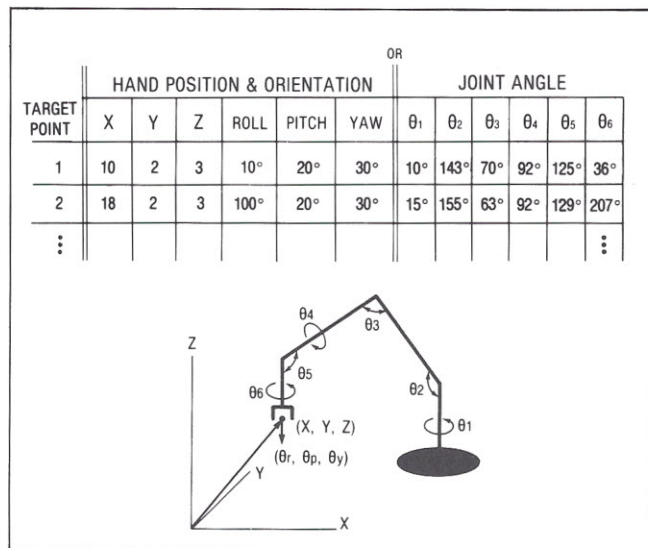


Figure 1. Two alternative ways to represent target points for a robot manipulator. On the left, the targets are defined by their coordinates in an external frame. On the right is their representation in terms of the joint angles of the arm. If the machine has any linear joints, replace the appropriate " $\theta_i$ " by an " $r_i$ ."

Another way of identifying a target to the robot is to give its location in some external coordinate system. This can be especially useful if the target's location is measured by vision or some other sensory system. It is also an integral feature of many high-level robot programming languages.

For the robot to use such a description, however, it must be able to *transform* a location given in external coordinates into the manipulator joint angles that correspond to that target. (I'll use the terms *joint angles* or *joint displacements* interchangeably to refer to the complete set of joint angles and/or linear displacements that uniquely characterize a given arm configuration.) Similarly, the robot may need to know, given a set of joint angles, where the arm is in that external frame.

The equations defining these relationships are referred to as the *kinematic solution* of a robot arm. A robot's kinematics describe its geometrical relationship with its environment. Since many of these relationships may involve angular rotations, both of the robot's joints and of the axes of the external frame, the transformations naturally require the evaluation of trig functions. A real-time robot control program needs a fast, accurate method of computing trig functions, especially if the transformations are included in the servo loops. In this article, I'll describe an efficient approach to trig computations that uses a combination of table lookup and linear interpolation.

Consider a robot manipulator with the cylindrical geometry shown in Figure 2. The location of the robot's wrist can be determined by measuring  $z$ ,  $r$ , and  $\theta$ , directly. Additional angles describe the orientation of the hand about that point, as shown in the figure. To find the location of the wrist in the Cartesian frame at the base of the robot, you must use the transformation:

$$x = r \cos \theta, y = r \sin \theta, z = z$$

Conversely, if you want to find the joint displacements that correspond to a given wrist location in that frame, you

must use the reverse kinematic solution:

$$r = \sqrt{x^2 + y^2}, \theta = \arctan(y, x), z = z.$$

The arctangent includes both  $x$  and  $y$  as arguments so that the appropriate quadrant is determined. The square root looks time consuming, but if you find  $\theta$  first, then you can make use of the fact that  $r = x \cos \theta = y \sin \theta$  to eliminate it. (I obtained this relation for  $r$  by rotating the  $x$  and  $y$  axes through the angle  $\theta$ .) Since we can easily compute both the tangent and cotangent from  $x$  and  $y$ , we can factor this relation for  $r$  to make the calculation easier:

$$r = \cos \theta (x + y \tan \theta) = \cos \theta (x + \frac{y^2}{x}), x \neq 0.$$

$$r = \sin \theta (x \cot \theta + y) = \sin \theta (\frac{x^2}{y} + y), y \neq 0.$$

The fastest way to evaluate trig functions is simply to use the input argument (angular measure) as an index which points directly to the correct value. Unfortunately, if the angular resolution is high, this table can grow extremely large. If you measure angles with 16 bits, for example, the table must have 65,536 separate values! An alternative approach used by most mainframe computers is to evaluate polynomials which correspond to truncated power series. This approach is compact because it uses no lookup tables; but obtaining sufficient accuracy requires evaluating several terms of the series, each with two floating point multiples and an add. On small computer, this may be time consuming, and error propagation can become a problem.

The approach I will describe offers a reasonable tradeoff between accuracy, memory requirements, and speed. The method is so effective that it is used by researchers and industry in some of the most sophisticated robot systems.

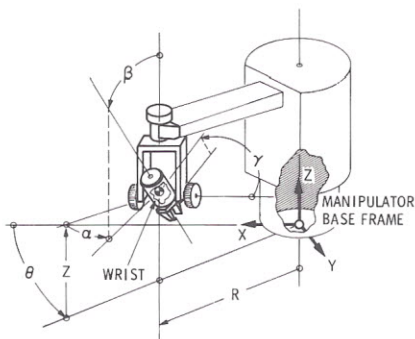


Figure 2. A robot with cylindrical geometry. The robot controls the  $r$ ,  $\theta$ , and  $z$  axes directly, as well as the hand angles. It must use kinematic transformations to relate measurements of  $r$ ,  $\theta$ , and  $z$  to the wrist position in the  $xyz$  frame at the base of the robot.

## Linear Interpolation

The basic idea of linear interpolation is to approximate the function over an interval by passing a straight line segment between the known values of the function at the endpoints of the interval, as shown in Figure 3. If we know  $f(x_1)$  and  $f(x_2)$ , and  $x$  lies between  $x_1$  and  $x_2$ , then:

$$f(x) \approx f^*(x) = f(x_1) + (x - x_1) \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

To get a complete description of the function over a larger interval or to increase accuracy, we can approximate the function by a series of such segments, as shown in Figure 4. We can simplify the evaluation of this by storing for each interval the *divided differences*:

$$D(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

Using these values in addition to the stored values of the function, we can compute the approximated function as:

$$f^*(x) = f(x_i) + (x - x_i) D(x_i),$$

for  $x$  in the interval between  $x_i$  and  $x_{i+1}$ . The calculation has only one add and one multiply, so it can be performed very rapidly.

This approach works nicely for functions which are "well behaved" (finite, continuous, etc.) in the interval of interest. Trigonometric functions, except for tangent near  $90^\circ$  and cotangent near zero, fall into this category. By using trig identities we can handle the exceptions as well.

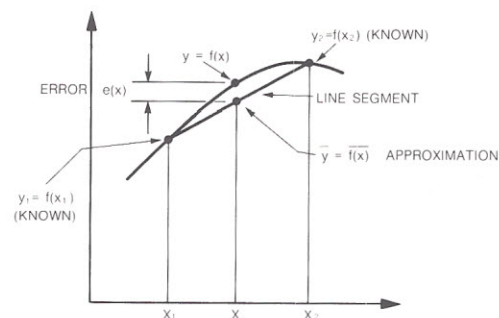


Figure 3. Linear Interpolation. The function  $f(x)$  is approximated over the interval from  $x_1$  to  $x_2$  by assuming that the approximate value lies on the line segment connecting  $(x_1, y_1)$  to  $(x_2, y_2)$ . This results in an error  $e(x)$  which can be made as small as needed by choosing  $x_1$  and  $x_2$  sufficiently close together.

## Reducing the Error

Purists may be concerned that the results of the interpolation are approximate. When you consider that the position feedback has only finite accuracy and that the

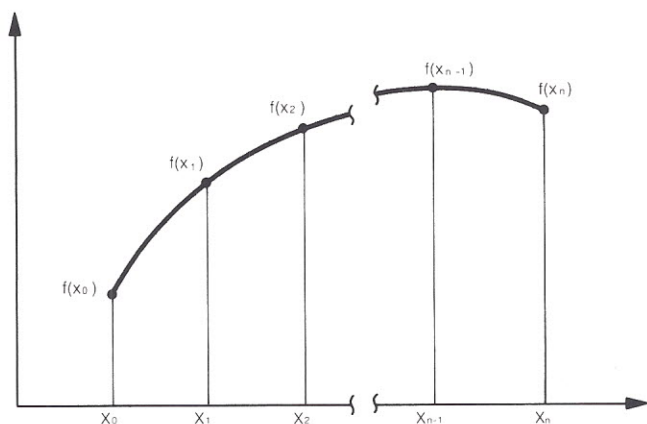


Figure 4. A function can be accurately approximated over a large range by linking a series of interpolation intervals. The first step is finding the interval  $i$  from  $x_i$  to  $x_{i+1}$  that contains  $x$  and then applying the linear interpolation procedure.

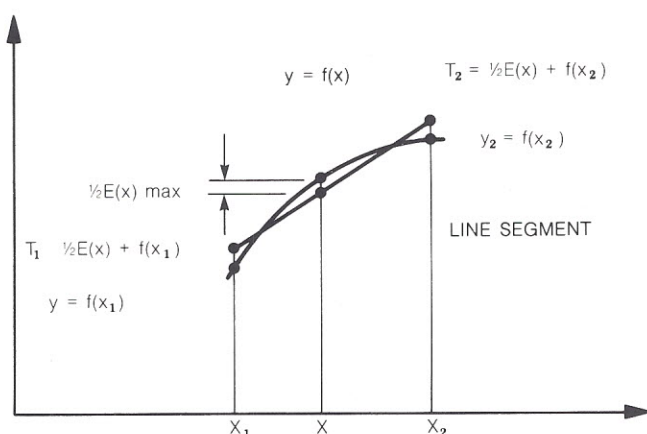


Figure 5. The maximum error in an interval can be cut in half simply by adjusting the table values  $T_1$  and  $T_2$  upward by one-half the maximum error. This distributes the error over the region.

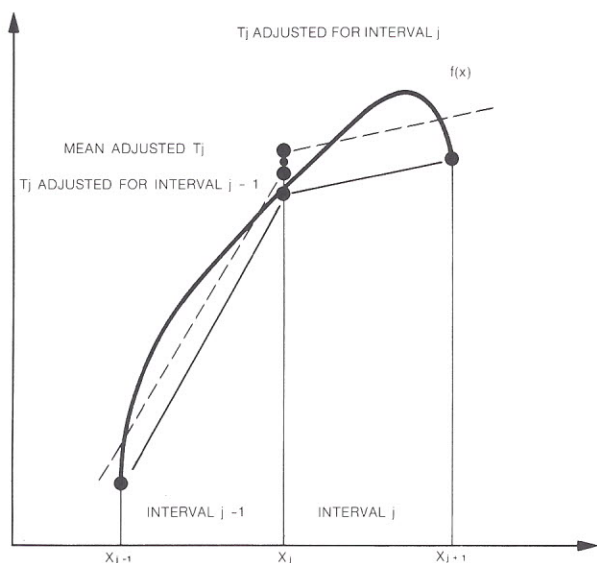


Figure 6. The mismatch caused by adjusting table values can be eliminated by setting the table value to the average of the adjusted values computed for the adjacent intervals.

control computer has only a finite word length, it's easy to show that these worries are unfounded. It is possible to select reasonable sizes for the interpolation intervals that will preserve all the accuracy available. It is pointless to maintain precision that cannot be used.

There are two sources of error in the approximation: that due to the rounding of arithmetic results and that due to the interpolation. The biggest contribution to rounding errors is caused when the divided difference  $D(x_i)$  is multiplied by the interpolation distance  $x - x_i$ . If we assume for the moment that we're using fixed-point arithmetic with rounded  $n$ -bit fractions, each table value will have an error bound of  $\pm 2^{-n-1}$ . The total error due to rounding will be approximately  $\pm 2^{-n-1} [2 + (x - x_i)]$ .

The interpolation error,  $f(x) - f^*(x)$ , depends on the shape of the function over the particular interval. We can find the point on the interval with the greatest interpolation error by differentiating the expression for the error with respect to  $x$ . The point with the most error is where the slope of the actual function is equal to the slope of the approximating segment. This is a useful result that we can use to reduce the interpolation error.

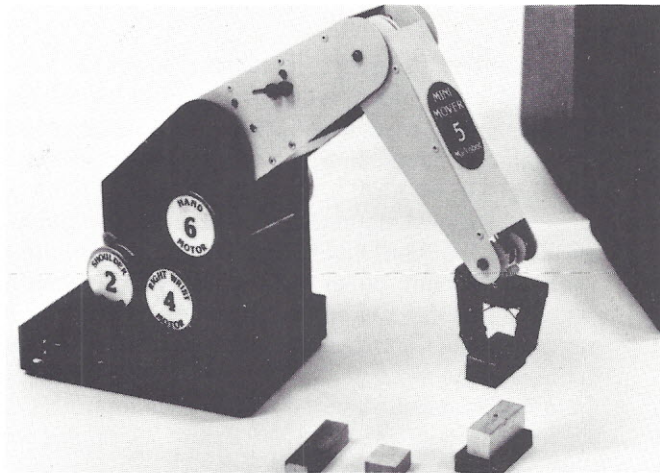
Another useful result of a more complete error analysis is finding the interval that has the greatest interpolation error. As you might expect, the interval with the greatest error contains the point where the function has the greatest curvature. In mathematical terms, the second derivative of the function corresponds to curvature, so that the curvature is greatest when the third derivative of the function is zero. This fact is useful later in choosing the width of the interpolation interval. In the interval that covers the region of peak curvature, we can express the maximum interpolation error as a function of the width of the interval and then find the width that results in an acceptable bound for the error.

The maximum interpolation error in an interval can be cut in half (at the expense of increasing the error at the endpoints of the interval) simply by raising the line segment by half the maximum error, as shown in Figure 5. This spreads the error over the interval, but it creates a mismatch between table values on adjacent intervals that could result in discontinuities in the approximation. We can handle this by adjusting a table entry by half the average of the error maximums of the adjacent intervals (see Figure 6), so that

$$T_j = f(x_j) + \frac{1}{4} [e_j(\max) + e_{j+1}(\max)].$$

This won't work on every interval, however, since limit conditions like  $\sin(\theta)=0$  have special significance for many kinematic calculations. On such intervals, just adjust one of the endpoints.

# Low Cost Table Top Robot Arm



## The Microbot MiniMover-5

The MiniMover-5 is a five jointed arm with a lifting capacity of 8 ounces when fully extended. Controlled by stepper motors, it has .013 inch (worst-case) resolution, allowing PRECISE CONTROL. The gripper end point may be positioned in the front half of a sphere with a radius of 17.5 inches. Speed ranges from 2 to 6 inches per second, depending upon the weight of the handled object.

- CABLE CONTROLLED from stepper motors in the body. This allows the MiniMover-5 to move faster and provide greater lifting capacity.
- UNIQUE DESIGN combining features from industrial robots and remote manipulators used in severe environments. This permits a lower cost by an order of magnitude.
- QUALITY CONSTRUCTION ensures long life and easy maintenance.



P.O. Box 17329, Irvine, Calif. 92713

New Toll Free Number, for order desk  
only (800) 854-8230.  
All other calls (714) 558-8813

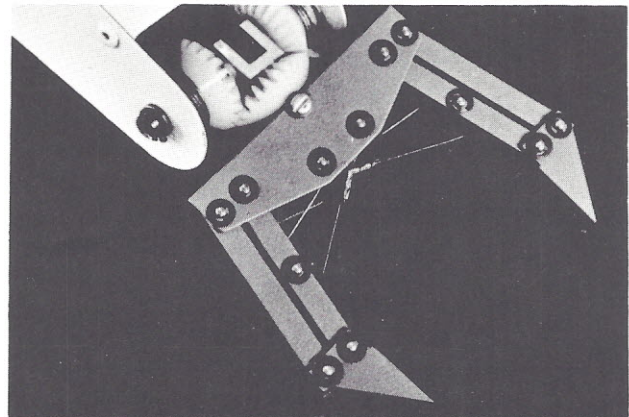
#### Perfect for:

- Evaluation of robots for industrial applications.
- Education in robotics principals and control.
- Experiments with artificial intelligence and robotics.
- Computer art and games.

Assembled as above

**ONLY \$1695**

100 Page Reference and  
Application Manual \$16.95



## Making the Calculations Fast

We've seen how the trig computations can be reduced to a table lookup, a subtraction, a multiplication, and an addition. We can speed up the calculation even more by a clever way of representing angles in the computer, one that proves much better than either degrees or radians.

The first step is to use a unit of angular measure such that the total number of units in a circle is an integral power of two. This is easy if the robot uses optical encoders for position feedback, since the number of pulses or states per revolution is usually a power of two. Similarly, we choose interpolation intervals that partition a quadrant of the circle into a number of intervals also equal to (a lesser) power of two. This yields several simplifications. The quadrant number, the number of the interval within the quadrant, and the interpolation displacement ( $x=x_i$ ) can all be directly masked out of the angular argument by logical operations. This amounts to using *turns* and *binary fractions* of a turn to represent angles, where one turn equals 360 degrees or  $2\pi$  radians.

If turns are stored as two's-complement fixed point numbers with the integer representing the number of full

turns and the fractions representing a partial turn, the fractional part has the familiar properties of angular displacements—a positive  $\frac{3}{4}$  turn and a negative  $\frac{1}{4}$  turn, for example, have identical representations, as do a full turn and a zero turn.

In this representation, the two most significant bits (MSBs) of the fractional part give the quadrant the angle lies in, and the next several bits identify which interpolation interval the angle lies in within that quadrant. The exact number of bits will depend upon the number of intervals in the table. The remaining least significant bits (LSBs) give the *absolute* displacement of the angular argument within the interpolation interval. (See Figure 7.) All this follows from making the number of interpolation intervals in a quadrant equal to a power of two. Since the trig functions are periodic, you only have to store lookup tables for one quadrant instead of four, and then use trig identities to evaluate the function in the other three quadrants.

Another feature of this approach is that it is easy to turn the absolute displacement of the angular argument (the LSBs of the fractional turn) into the *fractional displacement* of the angle within the interval,  $(x-x_i)/(x_{i+1}-x_i)$ , simply by a shift operation. All that's necessary is to shift the

## VOICE NEWS

An Independent Newsletter Reporting On  
**Speech Synthesis  
and  
Speech Recognition**

**VOICE NEWS** Keeps You Current On:

- New Speech Synthesis Products
- New Speech Recognition Products
- New Firms in the Speech Field
- Applications
- Voice Technology Services
- Conferences and Exhibitions
- Courses and Seminars

PLUS • Reports on the latest developments in the fast moving Voice Technology Industry

PLUS • **VOICE NEWS** provides name and telephone number contact information that puts you in touch with the industry's resources

One-Year Subscription (10 Issues): \$95.00

**Stoneridge Technical Services**  
P.O. Box 1891, Rockville, MD 20850  
(301) 424-0114

CIRCLE 29

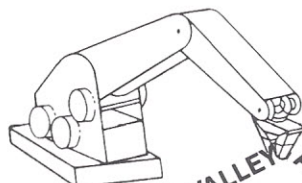


## ROBOTICS DEVELOPMENT

Robotic Arms for industrial, scientific and research applications.

Experience in applying the latest in micro-computer technology to San Diego's industrial and business problems since 1974.

- Develop your ROBOTICS capability
- Expand your Education and Training
- Economical and Portable
-  **apple II** and **TRS 80** interface
- Call on our experience — 452-0101



**\$1,995.00**  
Complete with  
Interface

**SORRENTO VALLEY ASSOCIATES, INC.**  
11722 Sorrento Valley Rd., San Diego, CA 92121  
CIRCLE 27

displacement field so that its MSB moves to the MSB of the fixed-point fraction (that is, the bit corresponding to  $\frac{1}{2}$ ). The resulting fraction equals the fractional displacement. This lets you store in the lookup table just the (adjusted) difference of the function across the interval instead of the divided difference  $D(x_i)$  mentioned earlier.

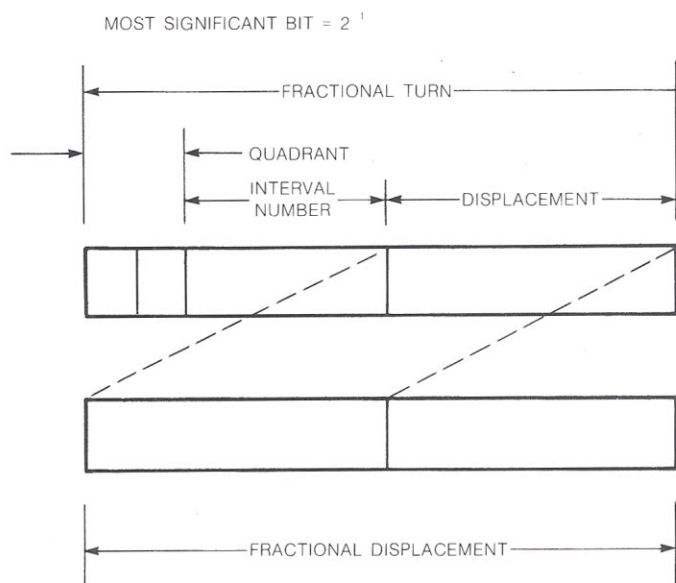


Figure 7. The quadrant, interval number, and the displacement of the argument within the interval can be extracted directly from the "turn" representation of an angle, provided that the number of interpolation intervals in the quadrant equals an integral power of two. The displacement of the argument can be converted to a fractional displacement simply by a shift operation.

## Evaluating the Trig Functions

### Sine(x)

To evaluate the sine function using a table that covers the first quadrant, use the following identities: (remember that the units are *turns*, so that  $\frac{1}{4}$  corresponds to  $90^\circ$ )

$$\sin(-x) = -\sin x, \text{ and}$$

$$\sin(\frac{1}{2} - x) = \sin x$$

To find sine values in quadrants 3 and 4, negate the argument, calculate the sine in quadrant 1 or 2, and negate the result. Use the second relation to find sine values in the second quadrant—subtract the argument from  $\frac{1}{2}$  and use the result as the argument to evaluate the sine in quadrant 1. Figure 8 shows these symmetry relations.

The sine function will have the greatest interpolation

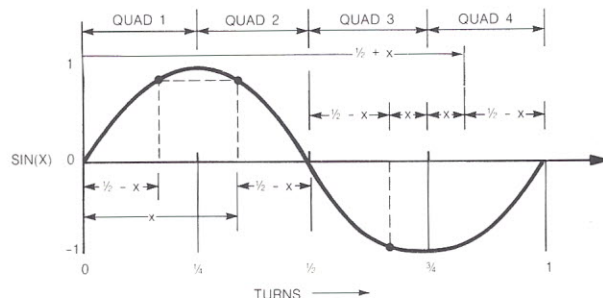


Figure 8. The sine curve. The curve in quadrant 2 is the reflection of that for quadrant 1 about the line  $x = \frac{1}{4}$  turn.

error when  $x$  equals  $\frac{1}{4}$  or  $\frac{3}{4}$  ( $90^\circ$  and  $270^\circ$ ). We can estimate this error as a function of the interval width,  $d$ , by supposing that the interval is centered about  $x = \frac{1}{4}$ , as shown in Figure 9. Remembering that the table values can be adjusted to cut the maximum error in half, we can say that:

$$2e = 1 - \sin(\frac{1}{4} - \frac{d}{2}) = 1 - \cos(\frac{d}{2}), \text{ or}$$

$$d = 2\arccos(1 - 2e)$$

If we want the interpolation error to be about the same as the error in representing the number  $\pm 2^{-n-1}$ , as described earlier, choose an interval width of:

$$d = 2\arccos(1 - 2^{-n})$$

To find  $N$ , the number of intervals in a quadrant, calculate:

$$N = \frac{1}{4d}$$

which is unlikely to be an integral power of two. Just choose the nearest power of two that gives sufficient accuracy—use the expression for  $e$  above to check it. Also, remember that to find the total error in the function evaluation you must add the rounding error as well.

### Cosine(x)

Since  $\cos(x) = \sin(\frac{1}{4} + x)$ , (arguments in turns, again) you can get cosine values by adding  $\frac{1}{4}$  to the argument and calling the sine routine.

### Tangent(x) and Cotangent(x)

Just use the defining relations:

$$\tan x = \sin x / \cos x \text{ and } \cot x = \cos x / \sin x.$$

### Arctangent(y,x)

Our arctangent function needs two arguments (an input vector), so that it can tell which quadrant the input vector is in by examining the signs of  $x$  and  $y$ :

$$y \geq 0, x \geq 0: \text{quadrant 1}$$

$$y \geq 0, x < 0: \text{quadrant 2}$$

$$y < 0, x < 0: \text{quadrant 3}$$

$$y < 0, x \geq 0: \text{quadrant 4.}$$

Save the quadrant number for later, then set the sign of each to plus to place the angle in the first quadrant. Next, find the actual tangent of the angle to use for the function evaluation. Since tangent blows up whenever  $x$  is too small, we can first find the cotangent and then use the relation  $\cot \theta = \tan(\frac{1}{4} - \theta)$ .

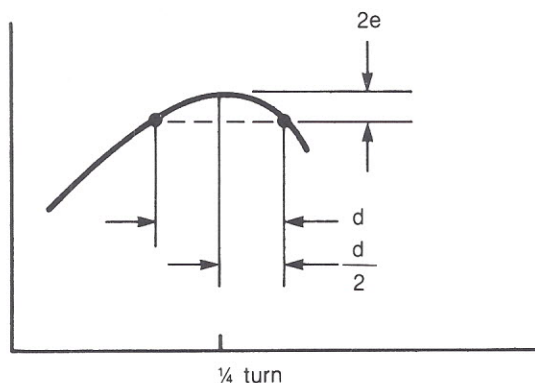


Figure 9. Estimating the maximum interpolation error sine.

A convenient way to program this is to choose the relation with the largest denominator: If  $x < y$  compute:

$$Q = y/x = \tan \theta.$$

Then find  $\theta = \arctan(Q)$  by interpolation from the lookup table. If  $x = y$ , then we know right away that  $\theta = 1/8$ , ( $45^\circ$ ) and we don't need any lookup or interpolation. If  $y > x$  then compute:

$$Q = x/y = \cot \theta.$$

Then  $\arctan \tan (1/4 - \theta) = \arctan (\cot \theta) = \arctan Q$ , so that  $\theta = 1/4 - \arctan Q$ .

Now we need to put  $\theta$  back into the correct quadrant. Using the quadrant number saved earlier, compute the output angle  $\theta$  to return as the result of the function:

- quadrant 1:  $= \theta$
- quadrant 2:  $= 1/2 - \theta$
- quadrant 3:  $= 1/2 + \theta$
- quadrant 4:  $= -\theta$ .

Since  $Q$  is always less than one, the interpolation argument is just a fixed-point fraction (that's why it's convenient to exclude the case  $x = y$ ). The interpolation region corresponds to the angles from zero to one count less than  $1/8$  turn.

We can perform an error analysis for the arctangent interpolation similar to what we used to find the interval size for the sine table. Arctangent has the greatest interpolation error for  $Q = 1/\sqrt{3} = .57735$ , which corresponds to  $1/12$  turn or  $30^\circ$ . By centering the interval  $d$  around this point and solving for  $d$  as a function of  $e$  like before, we get:  $d = \frac{8}{3} \sqrt{21e|\sqrt{3}|}$

Just as we did for the sine function, we must divide the interpolation region ( $0 \leq Q < 1$ ) into  $N$  intervals, where  $N$  is an integral power of two. Once you've picked an interval width, you can solve the above equation for  $e$  in terms of  $d$  and use it to verify your choice. Like before, the total error includes this interpolation error, plus the rounding error, and also, for arctangent, the error introduced by the division used to compute the tangent or cotangent.

## Using the Routines: An Example

Listing 1 shows a coding of the sine, cosine, and arctangent subroutines written in the MACRO-11 assembly language for the DEC PDP-11 series of computers.

Listing 1. The sine, cosine, and arctangent routines programmed for the PDP-11/34, using fixed point angle representation and floating point for the interpolation tables. Bit 15 is the MSB of a PDP-11 word.

```
.TITLE PGMATE
.ENABL LC
; Interpolation subroutines for trig functions
; and table reader to read in the tables from file DK:SINTAB.DAT
;
; latest version 22-OCT-80
;
; definition of the floating point registers
AC0=%0
AC1=%1
AC2=%2
AC3=%3
AC4=%4
;
; macro to push and pop registers on and off the stack
.MACRO PUSH A
MOV A, -(SP)
.ENDM
.MACRO POP A
MOV (SP)+, A
.ENDM
.PSECT TRIGRT
;
; The two word angle format is a 32 bit twos-complement number. Therefore
; the sin and cos routine ignore the first word which is number of complete
; turns. The second word is a fraction of a turn as follows:
; Bits 15-14 indicate quadrant. This is a value from 0 to 3 for the
; quadrants 1 through 4. Bits 13 through 8 are the table index
; having a range of 0 to 63. In octal word format, 1 part in this field
; has a value of 400 and the value ranges from 0 to 37400. Bits 7
; through 0 are the part used for interpolation between table values.
; The sine and cosine output and the arctan input are single precision
; floating point numbers in DEC format. This routine is callable from
; either a macro main program or a PASCAL main program and is in format
; compatible with the calling sequence used by OMSI (Oregon Software, Inc.)
; PASCAL.
;
; PASCAL callable as if declared:
; TYPE ANGLE = ARRAY[1..2] OF INTEGER;
; function PSIN(arg:ANGLE):real;
; function PCOS(arg:ANGLE):real;
; function PARCTN(sin,cos:real):real;
; Note that parctn returns a thing declared real which is really an angle
; in the two word angle format. The main program declarations will need
; a type union (variant record) to accomodate this.
; The Macro callable interface uses different names: TSIN, TCOS, TARCTN.
; There are no FP registers saved, others are. Call for TSIN and TCOS:
;
; Mov ^angle, R0      (^angle means a pointer to the angle)
; Mov ^result, R1
; JSR PC,TSIN or TCOS
;
; Call for TARCTN:
; Mov ^sin, R0
; Mov ^cos, R1
; Mov ^result, R2
; JSR PC,TARCTN
;
; .PAGE
;
; COSINE ROUTINE (COS(X) = SIN(90+X) with X MOD 360)
; .GLOBL TCOS,PCOS
; .GLOBL $B74,$B76
; $B74 and $B76 are routines in the PASCAL runtime system which save the
; registers and other context variables.
;
; .ENABL LSB
PCOS: JSR R0,$B74
MOV 36(6),R0 ;move the argument to R0
MOV #-1,-(SP) ;indicate that this was a PASCAL call
BR 1$
TCOS: PUSH R0 ;save some registers
PUSH R2
MOV 2(R0),R0 ;get the argument from the pointer
CLR -(SP) ;indicate that this was a MACRO call
1$: ADD #40000,R0 ;add 90 DEGREES to argument
BR FMCOS ;and jump to sine routine
.DSABL LSB
;
; SINE ROUTINE
; .GLOBL TSIN,PSIN
PSIN: JSR R0,$B74
MOV 36(6),R0 ;get the argument (word 2 only used)
```

```

MOV #1,-(SP) ;a PASCAL call
TSIN: BR FMCOS
PUSH R0 ;save registers
PUSH R2
MOV 2(R0),R0 ;get argument
CLR -(SP) ;a MACRO call
FMCOS: MOV R0,R2 ;SAVE ANGLE
CLR R0 ;GET TABLE LOOKUP PART
ASL R0
ASL R0 ;GET RID OF QUADRANT INFO
BCC 1$ ;IF QUADRANT IS NOT 2 OR 3
NEG R0
BEQ 2$
NEGB R2 ;change sign of interpolation part
BEQ 1$
SUB #2000,R0 ;index table from other end if interp part
;is non zero
1$: SWAB R0
LDF TBL(R0),AC1 ;get table entry into floating pt acc
LDF DTBL(R0),AC2 ;get difference table entry
MOV R2,R0
BIC #177400,R0
LDCIF R0,AC3 ;interp part
MULF AC2,AC3 ;mult by difference table
ADDF AC3,AC1 ;add in table entry
4$: TST R2 ;is this quadrant 3 or 4?
BPL 3$ ;NO
NEGF AC1 ;change sign
3$: TST (SP)+ ;was this a PASCAL or a MACRO call
BPL 6$
STF AC1,40(SP) ;store argument allowing for stuff $B76 has
;put on the stack
JSR R0,$B76 ;restore calling context
MOV (SP),4(SP) ;fix the return address and the stack
ADD #4,SP
RTS PC ;the end
6$: STF AC1,(R1) ;MACRO return sequence
POP R2
POP R0
RTS PC ;the end
2$: NEGB R2
BNE 5$
LDF #040200,AC1 ;this is a DEC floating 1.0
BR 4$
;
; ARCTAN OF A/B
; The table gives the sign of the arguments and tangent for each quadrant
; QUADRANT DATA: (QUAD:SIN/COS,TAN) Q1:++,+ Q2:+-,- Q3:--,- Q4:-+,-
; For Q3, Q4 we return a negative angle.
;
.GLOBAL TARCTN,PARCTN
PARCTN: JSR R0,$B74 ;save things according to PASCAL
LDF 40(6),AC0 ;get arg A
LDF 34(6),AC1 ;get arg B
MOV #1,-(SP) ;indicate PASCAL call
BR CMTN ;to common routine
TARCTN: LDF (R0),AC0 ;load arg immediately to floating acc
LDF (R1),AC1 ;as above
PUSH R0 ;save registers
PUSH R1
PUSH R3
CLR -(SP) ;indicate MACRO call
CMTN: CLR R3 ;quadrant info stored here
CLR R1 ;used to indicate which 45 deg octant
MOV #40000,-(SP) ;hope this really saves time, it saves space.
; test for which quadrant we are in and for zero arguments.
TSTF AC0
CFCC ;put floating condition codes into PSW
BLT 1$ ;ARG A IS NEGATIVE
BGT 2$ ;ARG A IS POSITIVE
TSTF AC1 ;ARG A is zero
CFCC
BPL 10$ ;ARG B IS POSITIVE
73$: MOV #100000,R3 ;ANGLE IS 180 OR -180 (gives -180 for result)
10$: TST (SP)+ ;GET RID OF #40000
TST (SP)+ ;PASCAL/MACRO SWITCH
BEQ 12$ ;go to MACRO exit
CLR 44(SP) ;SET + ANGLE
TST R3 ;Q3,4 CAUSE NEGATIVE ANGLE
BGE 11$
COM 44(SP)
11$: MOV R3,46(SP) ;STORE RESULT WD 2
JSR R0,$B76 ;RETURN ADDRESS
MOV (SP),10(SP)
ADD #10,SP
RTS PC ;This is one exit from routine
;
12$: CLR (R2)
TST R3
BGE 13$
COM (R2)
13$: MOV R3,2(R2)
POP R3
POP R1
POP R0
RTS PC ;This is other exit from routine
; if you get beyond here arg A was not zero
1$: NEGF AC0 ;ABS OF A
MOV #100000,R3 ;THIS IS Q3 OR Q4
2$: TSTF AC1 ;now check ARG B
CFCC
BLT 3$ ;B IS NEGATIVE

```

```

BGT 4$ ;B IS POSITIVE
BIS (SP),R3 ;ANGLE IS 90 OR 270 (B=0)
BR 10$ ;set angle and exit
3$: NEGF AC1 ;ABS OF B
TST R3 ;Q2 OR Q3?
BMI 41$ ;Q3
BIS (SP),R3 ;Q2
BR 41$
4$: TST R3
BPL 41$
BIS (SP),R3
41$: CMFP AC0,AC1 ;which octant (use table as TAN or CTN?)
CFCC
BGT 5$ ;WE ARE IN OCTANT 2 MOD 90 DEGREES
BLT 6$
BIS #20000,R3 ;(ARG A = ARG B) 45 DEG. (or odd multiple)
BR 10$ ;set result and exit
5$: DEC R1 ;INDICATE OCTANT 2
DIVF AC0,AC1 ;GET RESULT INTO AC0 FOR NEXT OPERATION
STF AC1,AC0
BR 7$
6$: DIVF AC1,AC0 ;RESULT NOW IN AC0 FOR NEXT OPERATION
7$: MODF 8$,AC0 ;INITIAL AC0 IS LESS THAN 1
STCFI AC1,R0 ;AC0 OUGHT TO BE <64 OR THERE IS AN ERROR
ASL R0 ;convert to index for table
ASL R0
LDF TNTBL(R0),AC2 ;GET TABLE PART
LDF DTNTBL(R0),AC1 ;and difference table part
MULF AC1,AC0
ADDF AC2,AC0 ;sum table and interp value
STCFI AC0,R0 ;convert to integer for result angle
BIT (SP),R3 ;IN Q2 OR Q4 ?
BNE 71$ ;NO
TST R1 ;(check octant) CTN ?
BEQ 9$ ;NO - TAN IN Q1 Q3
BR 72$
71$: TST R1 ;CTN ?
BNE 9$ ;YES- CTN IN Q2 ,Q4
72$: TST R0 ;IS THE ANGLE TOO SMALL?
; the interpolation may lead to a zero angle. In some cases this angle
; is in a different quadrant than the one indicated by the stored
; quadrant information which is valid for non-zero angles.
BNE 91$ ;NO
BIT (SP),R3 ;AT 0 OR 180?
BEQ 91$ ;NO
BIT #100000,R3 ;DO WE WANT 180 DEGREES
BEQ 73$ ;YES
CLR R3 ;NO, CLEAR EXTRANEIOUS BITS
BR 10$ ;R0 AND R3 CLEAR
91$: NEG R0
ADD (SP),R0
9$: BIS R0,R3
BR 10$
8$: .FLT2 64 ;a floating point 64
;
; TABLE LOADER PROCEDURE
; This procedure with no arguments is callable by TABLRE (or TABLREAD)
; in PASCAL. Dont use it with MACRO - its sets up its own stack which will
; probably overlap the normal MACRO stack and lose badly
; It can be used with either v1.1 or v1.2
; Destroys R0,R1
.GLOBAL RTAREA ;external in the PASCAL runtimes
.GLOBAL TABLRE ;internal in this routine
.MCALL .LOOKUP,.READW,.CLOSE,.PRINT,.EXIT,.WAIT
;
TABLE: MOV #5,R1 ;Only looks at first 6 channels
1$: .WAIT R1 ;is this RT-11 channel available?
BCS 2$ ;we got one
DEC R1
BPL 1$
3$: .PRINT #MSG1 ;all errors print same message
.EXIT ;and abort the program
; Dont reserve channel since we close it before exiting routine
2$: MOV SP,SPSV ;save stack pointer
;the pascal stack is at top of core and the USR will swap this part out
MOV #1000,SP
.LOOKUP #RTAREA,R1,#BLK
BCS 3$ ;error in finding file
.READW #RTAREA,R1,#DTBL,#256,#0 ;error in reading file
BCS 3$
.READW #RTAREA,R1,#DTNTBL,#256,#1
BCS 3$
; we should also check number of words read but dont do so
.CLOSE R1 ;release channel
MOV SPSV,SP ;restore PASCAL stack pointer
RTS PC
.NLIST BEX
SPSV: .WORD 0
FBLK: .RAD50/DK SINTABDAT/ ;name of file to read for the data
MSG1: .ASCIZ/? error reading SINTAB.DAT/
.EVEN
;
; KEEP FOLLOWING SIZES AND ORDER THE SAME. Data is stored here from the file
.GLOBAL DTBL
DTBL: .BLKW 128.
TBL: .BLKW 128.
DTNTBL: .BLKW 128.
TNTBL: .BLKW 128.
;
.PSECT
.END

```

The routines assume that the main program has loaded the function value and difference tables into memory. The subroutine `TABLEREAD`, at the end of the listing, shows an example of how to read a binary data file containing the tables to load them into their proper locations. For both the sine and arctangent tables, we used 64 interpolation intervals. The sine table, Table 1, lists the adjusted values of sine in the first quadrant in steps of  $1/256$  of a turn, together with the corresponding differences. The arctangent table, Table 2, shows the adjusted arctangent values and their differences.

## Conclusions

The methods I've described here are extremely useful for evaluating trig functions, and can be applied with little difficulty on any type of computer. You can also adapt this method of linear interpolation to other functions beside

Table 1. Sine values and differences for the first quadrant, with an interval of size of  $1/64$  of a quadrant.

1	0.0000000	0.0000959	33	0.7071338	0.0000670
2	0.0245426	0.0000958	34	0.7242748	0.0000653
3	0.0490700	0.0000957	35	0.7409793	0.0000635
4	0.0735678	0.0000955	36	0.7572377	0.0000611
5	0.0980213	0.0000953	37	0.7730399	0.0000599
6	0.1224158	0.0000950	38	0.7883765	0.0000581
7	0.1467365	0.0000947	39	0.8032380	0.0000562
8	0.1709698	0.0000943	40	0.8176159	0.0000542
9	0.1950981	0.0000938	41	0.8315012	0.0000523
10	0.2191099	0.0000933	42	0.8448856	0.0000503
11	0.2429898	0.0000927	43	0.8577612	0.0000483
12	0.2667233	0.0000921	44	0.8701200	0.0000462
13	0.2902960	0.0000914	45	0.8819547	0.0000442
14	0.3136940	0.0000907	46	0.8932582	0.0000421
15	0.3369030	0.0000899	47	0.9040235	0.0000399
16	0.3599090	0.0000890	48	0.9142445	0.0000378
17	0.3826983	0.0000881	49	0.9239145	0.0000356
18	0.4052570	0.0000872	50	0.9330282	0.0000334
19	0.4275716	0.0000862	51	0.9415798	0.0000312
20	0.4496287	0.0000851	52	0.9495642	0.0000290
21	0.4714149	0.0000840	53	0.9569765	0.0000267
22	0.4929172	0.0000828	54	0.9638125	0.0000244
23	0.5141225	0.0000816	55	0.9700679	0.0000222
24	0.5350182	0.0000804	56	0.9757390	0.0000199
25	0.5555916	0.0000791	57	0.9808223	0.0000175
26	0.5758303	0.0000777	58	0.9853149	0.0000152
27	0.5957221	0.0000763	59	0.9892139	0.0000129
28	0.6152552	0.0000749	60	0.9925170	0.0000106
29	0.6344176	0.0000734	61	0.9952223	0.0000082
30	0.6531978	0.0000718	62	0.9973281	0.0000059
31	0.6715846	0.0000702	63	0.9988331	0.0000035
32	0.6895669	0.0000686	64	0.9997365	0.0000010

Table 2. Values of arctangent and its differences over 64 equal intervals in the range 0-1. The first two columns of turns and differences are given as a normalized fixed-point octal fraction, and the second two columns are the actual decimal fraction values.

ARCTAN AND ARCTAN DIFFERENCE TABLE		NORMALIZED (OCTAL)		ACTUAL (DECIMAL)	
INDEX	TURNS	DIFF	TURNS	DIFF	
1	0	242	0.00000	0.00249	
2	242	242	0.00249	0.00249	
3	505	242	0.00497	0.00248	
4	750	242	0.00746	0.00248	
5	1213	242	0.00993	0.00247	
6	1455	241	0.01241	0.00247	
7	1717	241	0.01488	0.00246	
8	2160	240	0.01734	0.00245	
9	2421	240	0.01979	0.00244	
10	2681	237	0.02224	0.00243	
11	3120	236	0.02467	0.00242	
12	3357	235	0.02709	0.00241	
13	3615	234	0.02950	0.00240	
14	4052	234	0.03190	0.00238	
15	4306	233	0.03428	0.00237	
16	4541	231	0.03664	0.00235	
17	4773	230	0.03899	0.00233	
18	5224	227	0.04132	0.00231	
19	5453	226	0.04364	0.00230	
20	5702	225	0.04593	0.00228	
21	6127	223	0.04821	0.00226	
22	6353	222	0.05046	0.00223	
23	6575	221	0.05270	0.00221	
24	7016	217	0.05491	0.00219	
25	7236	216	0.05710	0.00217	
26	7454	214	0.05927	0.00215	
27	7671	213	0.06142	0.00212	
28	10104	211	0.06354	0.00210	
29	10315	210	0.06564	0.00208	
30	10525	206	0.06772	0.00205	
31	10734	204	0.06977	0.00203	
32	11141	203	0.07179	0.00200	
33	11344	201	0.07379	0.00198	
34	11545	177	0.07577	0.00195	
35	11745	176	0.07772	0.00193	
36	12143	174	0.07965	0.00190	
37	12340	172	0.08155	0.00188	
38	12533	171	0.08343	0.00185	
39	12724	167	0.08528	0.00183	
40	13114	166	0.08711	0.00180	
41	13302	164	0.08891	0.00178	
42	13466	162	0.09068	0.00175	
43	13651	161	0.09243	0.00173	
44	14032	157	0.09416	0.00170	
45	14212	155	0.09586	0.00168	
46	14370	154	0.09754	0.00165	
47	14544	152	0.09919	0.00163	
48	14717	151	0.10082	0.00160	
49	15070	147	0.10242	0.00158	
50	15237	145	0.10400	0.00156	
51	15405	144	0.10555	0.00153	
52	15552	142	0.10709	0.00151	
53	15715	141	0.10860	0.00149	
54	16056	137	0.11008	0.00146	
55	16216	136	0.11155	0.00144	
56	16354	135	0.11299	0.00142	
57	16511	133	0.11441	0.00140	
58	16645	132	0.11581	0.00138	
59	16777	130	0.11718	0.00135	
60	17130	127	0.11854	0.00133	
61	17257	126	0.11987	0.00131	
62	17405	124	0.12118	0.00129	
63	17532	123	0.12248	0.00127	
64	17656	121	0.12375	0.00125	

CNV=020000/(PI/4) IS:

CNV2=0.125/020000 IS:

10430.3779297

0.00001526

trig. I've used this method for trig calculations in two robotics projects with very successful results. In one case, using an IS-1000 minicomputer with 16 bit fixed-point arithmetic, we were able to calculate 15 bit sines and cosines in just 35 microseconds without even using difference tables. This let us control *two* robot arms and still have compute power left over for vision functions and extensive geometric calculations. Using these algorithms on your microcomputer will help make it possible for your robot to do fast coordinate transforms and attain higher levels of intelligent behavior.

*Acknowledgement: I wish to thank John Wedel of JPL for his work in programming these algorithms.*

## Help Wanted

MAJOR INDUSTRIAL COUNTRY needs 17,000 workers by 1985 to do dirty, demeaning, repetitious, and dangerous jobs. Applicants must be able to learn new tasks quickly and perform with a near-perfect level of accuracy. Breaks, lunch, health insurance, sleep, or wages beyond the subsistence level are out of the question. Inquiries: P.O. Box USA1

## The revised and expanded 2nd edition of INDUSTRIAL ROBOTS, FUNDAMENTALS, and INDUSTRIAL ROBOTS, APPLICATIONS will help your company fill the jobs nobody wants.

When you receive your copy of this new, two volume, 930 page, hardbound set you will discover it is one of the most comprehensive collections of technical material on industrial robots available.

You will explore entire chapters in Volume I, Fundamentals on:

- Implementation
- Justification
- Human Factors
- Capabilities
- Application
- Future Technology

In these chapters you will discover current, practical data on: robot decision making; safety equipment; employee and public relations; touch, sight, and remote center compliance; and much more.

Volume II, Applications details the methods manufacturers have used to successfully introduce industrial robots into virtually every process, including:

- Material Handling
- Machine Loading
- Press Loading
- Forging & Heat Treating
- Foundry
- Plastic Molding
- Welding
- Machining
- Assembly
- And More

In addition, you will find your revised and expanded 2-volume, 2nd Edition set contains detailed specs for many of the commercially available models of industrial robots. In a handy table you will explore complete information on the axes of motion, movement, end of arm speed, capacity, control, memory, programming method, potential applications and more.

Also, this new set is fully indexed, plus, contains a directory of robot manufacturers and distributors. Finally, hundreds of charts, tables, and photographs summarize and clarify complex points.



## Order Form



☐ Yes! I need the practical applications data in the 2nd Edition of Industrial Robots, Fundamentals, and Industrial Robots, Applications. Please send the indicated book (s). I have included \$1.00 per book for postage and handling.

	List	SME/RI Members
<input type="checkbox"/> Both Books	\$74.00	\$64.00
<input type="checkbox"/> Fundamentals (590)	\$37.00	\$32.00
<input type="checkbox"/> Applications (591)	\$37.00	\$32.00
SME/RI Member Number _____		<input type="checkbox"/> Non-Member

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City/State/Zip \_\_\_\_\_

- ☐ Payment Enclosed      ☐ Purchase Order Enclosed  
☐ Credit Card

Mastercard Account Number \_\_\_\_\_

Visa Account Number \_\_\_\_\_

Expiration Date \_\_\_\_\_

Signature \_\_\_\_\_

Send to:

Society of Manufacturing Engineers  
One SME Drive  
P.O. Box 930  
Dearborn, MI 48128  
Attn: Publication Sales

All prices subject to change without notice. All orders without proper member number will be billed at the non-member rate. Please remit in U.S. Funds. Allow 4-6 weeks for delivery.

By Jerry W. Saveriano

## AN INTERVIEW WITH

# GEORGE DEVOL

The inventor of the industrial robot  
talks about the creation of the industry and  
the forces now shaping its future.



*"Unimation got started back as early as 1956, when I met one George Devol at a cocktail party."*

So recalled Joseph Engelberger, the "father of industrial robots." Out of that meeting between entrepreneur, Joseph Engelberger, and inventor, George Devol, emerged a new industry—robotics. In an earlier interview (Jan/Feb 1981), we spoke to Joseph Engelberger. Now, we turn to George Devol, the inventor of the industrial robot.

George Devol was born in Louisville, Kentucky in 1912. In the 1930's, he was the first to apply photocells for controlling industrial machinery. The company he formed, United Cinephone, was the first industrial control firm

in the US.

Devol is primarily an inventor, and the list of his patents is long and impressive. He holds patents on photoelectric controls, on magnetic recording systems, on the first teachable machine, and on the first industrial robot. A package of patents, which he sold to Unimation, contains over 40 robotics-related inventions.

Devol has influenced the business world as well as the field of engineering. His expertise has helped Unimation become a leader in the robotics industry. And he continues to head his own firm, Devol Research Associates. In this interview, the fourth in the series, we look at industrial robotics through the eyes of George Devol.

*What led you to invent the industrial robot? How did you get started?*

Let's go way back, to around 1932. I had a different idea on variable area recording than RCA Photophone had. There were only two companies in that business then, RCA and Western Electric. They were big time; I was just a kid with an idea.

But my father had a little money, and I fooled around and came up with a product. Unfortunately, it wasn't any better than the products already on the market. I couldn't sell it.

Yet here I had photocells, used as pickups in projectors. I started looking around—there must be something I could do with photocells other than TV. I had the idea of making photoelectric controls for electric doors. And with patent in hand, I formed United Cinephone Corporation—as far as I know, the first industrial control company in the US.

One of our applications was controlling the registration on printing presses. Color printing presses made overlays of color. But sometimes the paper would shift. A fellow had to stand over the press, adjusting the advance and retard on the rolls to keep registration and colors right. We introduced photoelectric controls for this.

*But photoelectric controls didn't lead you to invent the Unimate.*

No. That line of development began later—in the 1940's. At that time, I had started thinking of ideas in magnetic recording. There were all kinds of AC recording techniques. So I thought: who's doing anything with DC recording? I looked around, and saw there was nobody. So, with men who have associated with me ever since, I applied for patents on DC magnetic recording. This is recording at rest, rather than in motion.

Next, we took our recording system and hooked it up to a machine—a lathe, for example. We turned out whatever parts we wanted and, in the process of making them, we magnetically recorded all the lathe's actions. From that point on, the lathe could automatically produce identical parts.

This was back in 1945 or 1946. We applied for a patent on a teachable machine. Nothing was close to that, except a digitally controlled machine built by Jay Forrester at MIT. But he was programming his machine, not teaching it. There's quite a difference.

*So the first Unimate was an extension of your teachable machine.*

Exactly. After we had our teachable machine, we thought: why not make a manipulator? Let's put a hand on the machine and move parts around. This was in the early 1950's. We applied for a patent on our manipulator—but we didn't call it a robot. We called it a Unimate. And as far as I know, it was the first patent that ever had the name of a product on it: "Unimate," from *Universal Automation*.

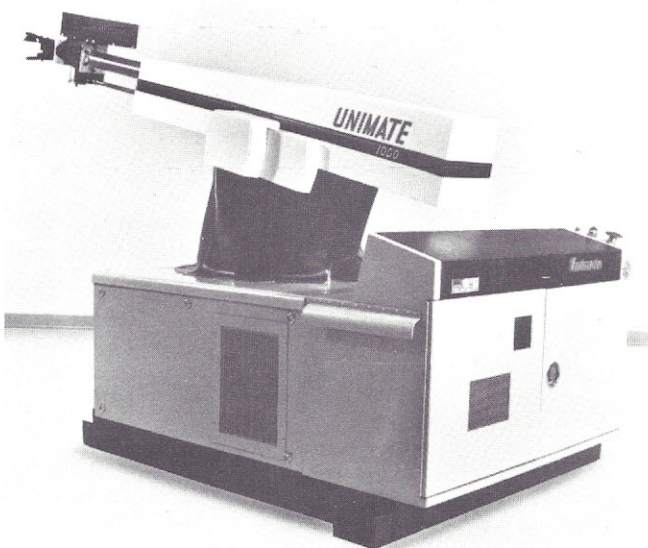
After 1954, with patent in hand, we tried to sell our idea. Tom Watson of IBM was interested. But at that time IBM was making its big push in computers. Their hands were full.

*Is that when you met Joseph Engelberger?*

Joe Engelberger was chief engineer of the Aircraft Products division of Manning, Maxwell, and Moore, near Bridgeport, Connecticut. I had talked to Harry Moore, who was Joe's boss at the time, and he steered me to Joe.

First, Joe and I talked about magnetic recording, the kind of business his company could easily get into. Since robots were an expensive proposition, we made our first contract on the basis of the magnetic recording patents. Manning, Maxwell, and Moore licensed the patents and agreed to do further development work.

In the meantime, Dressler Industries bought Manning, Maxwell, and Moore, and they didn't want any part of the Aircraft Products Division. But we believed in our ideas,



*The Unimate Series 1000 Industrial Robot.*

and went looking for someone else to buy the division.

After some searching, I found Norm Schafler, the president of Consolidated Diesel Electric Corporation (Condec). Condec agreed to buy the Aircraft Products Division, and Unimation was born.

---

**"We applied for a patent on our manipulator—but we didn't call it a robot. We called it a Unimate. And as far as I know, it was the first patent that ever had the name of a product on it: 'Unimate,' from Universal Automation."**

---

*This happened around 1958?*

Yes. In 1958, we decided to build a robot. By 1961, we had gotten our first Unimate out. I sold that machine myself—to General Motors.

*At that point, how did you view the Unimate? Did you say, "this is a robot," or did you see it as a manipulator or programmable teaching machine?*

It was a manipulator. Later on, everyone called it a robot. You should have seen the time I had with Joe, getting him to call it a robot. He said, "Don't do that. We'll never get in the automobile industry or anyplace else." I said, "Joe, look at the advertising business. If you want to sell something, you have to give it a name that everybody recognizes. No one knows what a manipulator is. Call it a robot." And we did.

*Joe Engelberger has the credit for being the father of industrial robots.*

He deserves an awful lot of credit. He was the one fellow who took the ball and carried it. But words are only words. You can call him "father." But I'm the crazy inventor of the robot. The patent office says so.

*Back in 1961, did you realize how important robots would be?*

Let's put it this way: I was concerned with building a better mousetrap, and this mousetrap was going to produce products for less money. That's how major economic change comes about. But I wasn't thinking of science fiction robots. I had to be practical, because the people I was selling to were practical. A production

engineer is the most hassled person on the face of the earth. Suppose you come and say, "We're going to put a robot in your plant." He'll say, "My God! I'll have 500 times more problems than I can ever solve." That's the kind of attitude we face.

*What did you do after the first Unimate was built?*

I kept getting more patents all the time. You see, I wasn't an employee of Unimation at any time. Along with my partner and patent attorney, Paul Martin, I ran my own patent business.

*Patents on Unimates?*

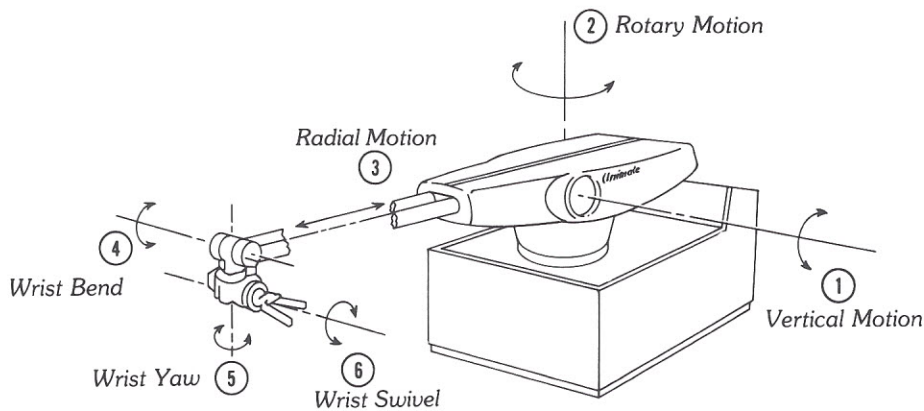
At that time, mostly on Unimates. We had quite a few patents. But it became evident that Condec was not a big enough company to carry on in Robotics without more capital. That's when Pullman, a large company, bought about 80% of Unimation. X million dollars later, Unimation had a whole product line of robots. Joe went overseas and started to sell them wholesale. In the US, people were still not very interested.

*When you say "overseas," do you mean Japan?*

---

**"I'm the crazy inventor of the robot.  
The patent office says so."**





Six program controlled articulations of Unimate industrial robot.

The Japanese licensed from Unimation. But I mean Europe. Unimation opened an office in England. And for a long time, a large percentage of Unimates were sold in Europe. Fiat was one of the biggest customers.

*Pullman lost a fortune on the Unimate for a long time.*

No. After four or five years, Pullman got discouraged—

*To the tune of 14 million dollars!*

They should have stuck it out! Nobody was kidding them. I told them: "Look, it's going to take five or ten years. Don't come in unless you're willing to spend a lot of money for that length of time." Pullman should have known that. They were smart enough to get into robotics; but they weren't smart enough to stay in long enough. I think they'll regret pulling out. Unimation still has cash flow problems, but they're on their way to becoming an extremely viable company.

---

**"Unimation bought a package of patents, and the rights up to X millions—and I won't go into the number, but I mean tens of millions—for the overall package."**

---

*Have you always been close to Unimation?*

Always. They're still willing to buy patents from me when I get them. Remember, though, that I'm a free agent. Unimation bought a package of patents, and the rights up to X millions—and I won't go into the number, but I mean tens of millions—for the overall package.

A patent is a long-term capital gain if you sell it. If you license it, it's regular income. This was an outright sale.

*How many patents were there?*

Some 40 odd.

*How do you see the robot business today?*

There are really three robot businesses now: There's the body business, the brain business, and the systems business. Cincinnati Milacron and Unimation are in the body business. Computer control is a big business in itself. Whatever computers have that robots can use, fine, but don't get them all mixed up together. As long as robots can talk to computers, that takes care of that. The systems business came out of the problem that most people in manufacturing plants don't know how to use robots.

---

**"A production engineer is the most hassled person on the face of the earth. Suppose you come and say, 'We're going to put a robot in your plant.' He'll say, 'My God! I'll have 500 times more problems than I can ever solve.'"**

---

The body business will, I think, always be the primary one. You don't have a system until you have the robot body. Whoever can build a flexible robot body at the right price—and which can be connected to various systems—is going to make a lot of money. The big volume is going to be in bodies, not systems.

*Do you class sensors as part of the brain business?*

Yes. And I think there are some interesting problems to be solved in this area. For instance, I'm interested in spatial control. We need something that will give the distance to and location of objects, 360° around, all the time, continuously. There's nothing I know of that can do this, and I mean do it all the time. I don't mean scanning around; I mean continuous spatial knowledge. This is the missing link in the robot as far as I'm concerned—much more so than vision.

*Yet vision is now getting the bulk of the attention.*

U.S. POSTAL SERVICE STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION (Required by 39 U.S.C. 3685)		
1. TITLE OF PUBLICATION Robotics Age		2. DATE OF FILING 9/25/81
3. FREQUENCY OF ISSUE BI-monthly		4. ANNUAL SUBSCRIPTION PRICE \$15.00
5. COMPLETE MAILING ADDRESS OF KNOWN OFFICE OF PUBLICATION (Street, City, County, State and ZIP Code) (Not printers) 10049 Commerce Ave., Tujunga, CA 91042		
6. COMPLETE MAILING ADDRESS OF THE HEADQUARTERS OR GENERAL BUSINESS OFFICES OF THE PUBLISHERS (Not printers) 10049 Commerce Ave., Tujunga, CA 91042		
7. FULL NAMES AND COMPLETE MAILING ADDRESS OF PUBLISHER, EDITOR, AND MANAGING EDITOR (This item MUST NOT be blank)		
PUBLISHER (Name and Complete Mailing Address) Phil Flora, 2521 Fairmont, Pear, La Crescenta, CA 91214		
EDITOR (Name and Complete Mailing Address) Alan Thompson, 3011 Community Ave., La Crescenta, CA 91214		
MANAGING EDITOR (Name and Complete Mailing Address) Joel Wilf, 1021 Harper #1, Los Angeles, CA		
8. OWNER (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) (Item must be completed)		
FULL NAME COMPLETE MAILING ADDRESS		
Alan Thompson 3011 Community Ave., La Crescenta, CA		
Phil Flora 2521 Fairmont, Pear, La Crescenta, CA		
Joel Wilf 1021 Harper #1, Los Angeles, CA		
9. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES (If there are none, so state)		
FULL NAME COMPLETE MAILING ADDRESS		
None		
10. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES (Section 411.3 GMM only) The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)		
(1) HAS NOT CHANGED DURING PRECEDING 12 MONTHS <input checked="" type="checkbox"/> (2) HAS CHANGED DURING PRECEDING 12 MONTHS <input type="checkbox"/> (If changed, publisher must submit explanation of change with this statement)		
11. EXTENT AND NATURE OF CIRCULATION		AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS
A. TOTAL NO. COPIES (Net Press Run)		11,167
B. PAID CIRCULATION 1. SALES THROUGH DEALERS AND CARRIERS, STREET VENDORS, AND COUNTER SALES 2. MAIL SUBSCRIPTION		3,406 8,061
C. TOTAL PAID CIRCULATION (Sum of B.1 and B.2)		11,529
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES		108 311
E. TOTAL DISTRIBUTION (Sum of C and D)		10,084 11,840
F. COPIES NOT DISTRIBUTED 1. OFFICE USE, LEFT OVER, UNACCOUNTED, SPOILED 2. RETURN FROM NEWS AGENTS		1,003 30 0
G. TOTAL (Sum of E, F.1 and 2. Should equal net press run shown in B)		11,167 13,000
12. I certify that the statements made by me above are correct and complete		
SIGNATURE AND TITLE OF PUBLISHER, BUSINESS MANAGER, OR OWNER <i>Phil Flora</i> Publisher		

PS Form 3526 June 1980 3128 (Page 1)

(See instruction on reverse)

We need vision, there's no question about it. But it doesn't answer all the problems. The worst problem right now is that if you take a servo-controlled arm and shoot it at an object at high speed, you want to know how close it is. The robot has to know when to stop the very second it touches the object. And it has to do this at tremendous speeds.

**"There are really three robot businesses now:  
There's the body business,  
the brain business,  
and the systems business."**

If you have an object made out of ferrous metal, you can use some kind of magnetic field detector. But the field of a magnetic detector is so wide that its definition is horrible. It spreads all over the place. You need something else, the X-thing I've been talking about. If this device were available—and simple enough—it would open up whole areas in the business that we can't touch today.

While we're on the subject of sensors, there's another point I should make: There are many things that people are designing complicated vision systems for—and some of them could be done much more simply.

Let's say, for example, that you have a flat piece of metal on a conveyor. Why not coat this metal, when it's made, one side light and one side dark. Immediately, with a simple photocell, you can get half the information you need. And it costs practically nothing to do this.

Why not put a precision locating handle on the sprue\* of a die casting machine, instead of a round knob. Once the robot grabs the handle, it can hold the part to an accuracy of five of six thousandths of an inch. Why go and mix up manufacturing parts, then have all this elaborate equipment to separate them?

**"It's perfectly all right for Japanese companies to have cartels; it's absolutely illegal for US companies. You have to say that the guy with the cartel is breaking the law in the United States, and his products can't come in."**

My principle has always been to reduce the complexity of the manufacturing system as much as possible. Even with a terribly difficult problem—springs, for example—there may be a simple solution. If you're making springs, why not coat each spring with a polyethylene film? Now, the damned things don't get intertwined anymore. They're nice little cylinders. You had to add something to the manufacturing process—but that something was inexpensive.

*It seems that the Japanese have been tackling these kinds of manufacturing problems. They seem to be in a better position to implement advanced robotic technology.*

One of the reasons is that their government is helping them, ours is not.

*Don't you think they have better manufacturing leadership? For a long time in the US, management didn't spend money on new equipment and techniques.*

The Japanese—and Germans, too—were forced to build new equipment because World War II destroyed the old. This didn't take a lot of brains, it took a set of circumstances. But frankly, I think people are going to

\*A sprue is an opening in a machine through which molten metal is poured.—Ed.

start shutting their doors on the Japanese. It's happening in Europe already. In France, there are thousands of Japanese cars still sitting at the docks, and they've been there for months.

*I'm not sure that's the right answer—protectionism.*

What are you going to do, let them ruin your economy? Let's not call it protectionism. Let's just say that until everybody plays by the same rules, we won't play the game. It's perfectly all right for Japanese companies to have cartels; it's absolutely illegal for US companies. You have to say that the guy with the cartel is breaking the law in the United States, and his products can't come in. Here we have rules and regulations that allow companies to do some things but not others. In Japan, companies can do a lot of things that we can't do—that's what it amounts to.

---

**"...when somebody builds a completely robotic plant, the guy across the street has to do the same—or go out of business."**

---

*You must admit, though, that the Japanese were smart enough to go ahead and implement advanced automation. They give their engineers the time and money it takes to do the job.*

True. But the Japanese don't have the same problem with stockholders and the bottom line that we do in this country. In Japan, the banks finance companies, not the general public, who finance American firms.

What does this mean? The fellow who invests in something doesn't want to sit and wait for six years to get something out of it again. Banks could do it, but the US government won't let them. There have been federal laws since the 1920's that stop banks from issuing stock for corporations and buying big interests in everything. This is a fundamental part of US law we're talking about.

Even so, robot factories are inevitable in this country. Companies will learn that if they build full plants of robots from the ground floor up, they'll have a lot less trouble than if they try to take a plant that has 10,000 people in it and move in 6,000 robots. And when somebody builds a completely robotic plant, the guy across the street has to do the same—or go out of business.

*What is your involvement with the robot industry today?*

I mostly work through Devol Research. It's a slightly new concept: A fellow comes to us with an idea, and we find out how he's going to make money, how we're going to make money, and how investors can invest in it so they're sure they won't lose any money. We set up individual patent groups. The group includes anybody that invests in the patent, the fellow who puts the idea in, and the one who physically works on it.

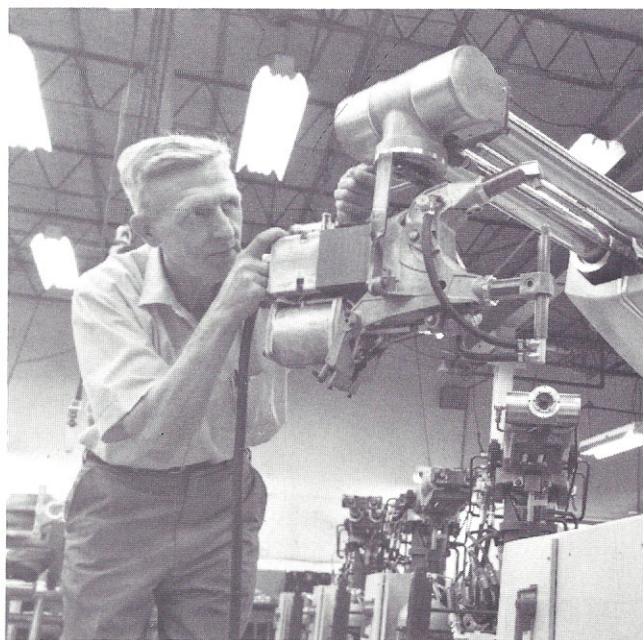
---

**"The only sensor I know of that really does a good job is the infrared sensor for a die cast machine."**

---

When something finally happens with that patent—when its sold, for example—it's a long-term capital gain. In the meantime, all the investors can deduct their costs per year against their regular income, just like a Small Business Investment Company (SBIC).

Devol Research has other advantages, too. The inventor doesn't have to sell his own idea any more; we have the marketing organization. An inventor working for us will be paid a reasonable salary, a stipend, but he's really going to make his money out of the patents he's creating for us. And each inventor will be part of his own little group. He won't have to worry about anything else. All we need is the



*Rigging the Unimate for spot welding.*

first guy who gets 10 million dollars a year income out of something, and we have it made.

*Do you think Devol Research will do more work in robot bodies or in, say, sensors?*

In sensors. It doesn't make sense for us to go into the body business. That's very capital intensive, and there are enough companies, right now, building robots.

But in sensors, there are a lot of missing links. The only sensor I know of that really does a good job is the infrared sensor for a die cast machine. That sensor works like a charm, finding out whether the sprue is in the machine. But we have some other good sensor patents in the works.

*In every interview, I ask the "human factors" question. How do we face the social problems caused by robotic technology?*

I often get asked that question. I try to steer clear of it, because it can cause an argument right away. And I don't know the answer. I don't think anybody does.

*I ask the question because you're one of the people who are bringing this technology about. Who should address these questions?*

I think that's up to the people we hire in Washington and elsewhere—the overseers of all social problems. Anything we can do to help them, I think we should do. But handling social problems is a different field than the one we're trained in. How can we address ourselves to something so far out of our areas of expertise?

---

**"...it's a lot easier to think big than small. For some crazy reason, the bigger an invention is, the easier it is to do it."**

---

*One final question: What advice would you give to a budding young inventor who wants to get into robotics?*

First of all, I hate to give advice. The best I can say is it's a lot easier to think big than small. For some crazy reason, the bigger an invention is, the easier it is to do it.

*Could you give us an example?*

The Unimate. If it had been just another piece of automation equipment, it might never have gotten anyplace. The Unimate took one hell of a lot of time, energy, money, and dedication. But if it worked, it promised to change the face of the world. On the other hand, if I had developed the best registration control in the world, how big a business could it be? Five, six million a year—you can't get anybody excited about that. An inventor's idea has to turn into some kind of product, and the bigger the product, the more demand there is for it. This makes it easier to sell the idea, and easier to find money to put into it.

*The small guy who's just starting out, he may have some good ideas, but he can't start out at that level. He has no credibility.*

He's going to come work for Devol Research. □

## For Robotics Research, Education, ... and Pure Enjoyment



### THE LOW-COST MINIMOVER-5 —\$1695. TABLETOP ROBOTIC ARM BY MICROBOT, INC.

#### USE IT AT —

- The Lab
- Classroom
- Commercial Environment—or simulate Industrial Robots
- Workshop—with your Personal Computer

#### BUY IT —

For the cost of a good peripheral

#### INTERFACES WITH —

Apple II\*, PMC-80,  
Radio Shack TRS-80  
(Model I/Level II)  
computers

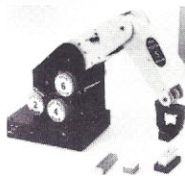
#### MICROBOT, INC. PRODUCTS DISTRIBUTED BY:

DORING ASSOCIATES  
1744 RTE. 9  
CLIFTON PARK, N.Y.  
12065-2497

(518) 371-9499



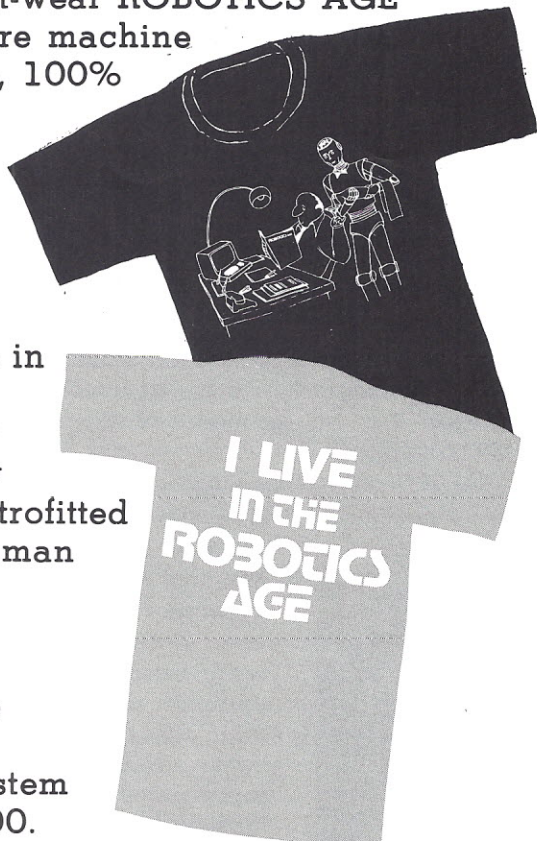
Eastern U.S.  
Distributor



\*Apple II is a trademark of Apple Computers, Inc.

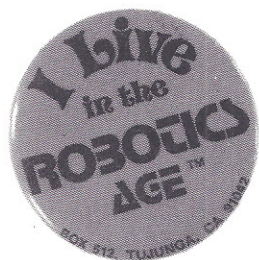
## Robotize Your Wardrobe!

These soft-wear ROBOTICS AGE T-shirts are machine washable, 100% cotton.



Available in maroon and navy blue, they can be retrofitted to any human male or female upper body. The complete T-shirt system costs \$7.00.

For those who insist on wearing metal, ROBOTICS AGE buttons are available for \$.50 each. On all orders, be sure to add \$1.00 for postage and handling.



NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

MEN'S	WOMEN'S	CHILD'S
_____ Small	_____ Small	_____ Small
_____ Medium	_____ Medium	_____ Medium
_____ Large	_____ Large	_____ Large

Buttons \_\_\_\_\_ @ \$.50 each

Be sure to add postage & handling

☐ Payment enclosed ☐ Bill my ☐ MasterCard ☐ VISA

Acct. # \_\_\_\_\_ Exp. \_\_\_\_\_

SIGNATURE \_\_\_\_\_

Send to: ROBOTICS AGE, POB 725, La Canada, CA 91011

## Back Issues

# ROBOTICS AGE

## Did You Miss Any?

**SUMMER 1979:** Digital Speed Control of DC Motors; Industrial Robotics '79; Introductions to Robot Vision; The Grivet Chess Playing Arm; Robotics in the Soviet Union; The Robotics Age Competitive Event.

**WINTER 1979:** Advances in Switched-Mode Power Conversion (Part I); Prospects for Robots in Space; Robotics Research in Japan; Report from IJCAI6.

**SPRING 1980:** Microcomputer Based Path Control; Robotics Research in France; Multiple Sensors for a Low-Cost Robot; The Robots of Autofact II; Inside Big Trak.

**SUMMER 1980:** Industrial Robots: Today and Tomorrow; Introducing the MiniMover 5; Advances in Switched Mode Power (Part II).

**FALL 1980:** Using Optical Shaft Encoders; Interview with Victor Scheinman; Robot Vision for Industry; The Autovision System; Industrial Robotics '80; Robots on Your Own Time: Superkim Meets ET-2.

**JAN/FEB 1981:** A Robot Arm Without a Budget; An Interview with Joseph Engelberger; Robots V—Dearborn 1980; Robots on Your Own Time; Opto "Whiskers"; Robot Toy Designs.

**MAR/APR 1981:** Video Signal Input; Chain-Code; Camera Geometry for Robot Vision; TIG Welding with Robots; Robot Digestive Tract—Robots On Your Own Time.

**MAY/JUNE 1981:** Rehabilitative Robots; A Homebuilt Computer Controlled Lathe; An Interview with Charlie Rosen; Superkim Meets ET-2; Part II.

**JULY/AUG 1981:** Segmenting Binary Images; The Robot as Transfer Device; Continuous Path Control of Stepper Motors; TIMEL: A Homebuilt Robot.

**SEP/OCT 1981:** Bullish Days in the Robot Business; Edge Detection in Man & Machine; Continuous Path Control with Stepping Motors; Build a Low-Cost Image Digitizer; Report from JACC-81; The Robot Builder's Bookshelf.

_____ Copies of SUMMER 1979	
_____ Signed, numbered Collector's Edition	<b>Sold Out!</b>
_____ Xerox copy of original	\$7.50/ea.
_____ Copies of WINTER 1979	\$2.50/ea.
_____ Copies of SPRING 1980	\$2.50/ea.
_____ Copies of SUMMER 1980	\$2.50/ea.
_____ Copies of FALL 1980	\$2.50/ea.
_____ Copies of JAN/FEB 1981	\$3.00/ea.
_____ Copies of MAR/APR 1981	\$3.00/ea.
_____ Copies of MAY/JUNE 1981	\$3.00/ea.
_____ Copies of JULY/AUG 1981	\$3.00/ea.
_____ Copies of SEP/OCT 1981	\$3.00/ea.

Send in your name, address, and payment to receive any back issues of ROBOTICS AGE you've missed. Payment must accompany order—back issues cannot be billed.

Add \$1.50 postage and handling for the first back issue, each additional back issue add 50¢.

Total Postage and Handling \_\_\_\_\_

Total Payment Enclosed \_\_\_\_\_

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

# THE GREAT JAPANESE ROBOT SHOW

by Joel M. Wilf

Affluent. That's how Tokyo feels. Cars on the street are solid and new. Goods in the shops are well-made. Department stores, seven floors high, are stocked with the latest stereos, video systems, cameras, and home computers. A country—whose GNP was once only 5% of America's—now has one of the highest standards of living in the world.

Robotics is one of the reasons. Japan has led the rest of the world in automating its factories. More than 77,000 robots work in their plants, more than 70% of the world's robot installations.

Despite this, most of us in the US know surprisingly little about Japanese robotics. Few of their machines have been sold in this country. (In 1980, only about 3% of Japan's industrial robots were exported.) And the reports we do get on Japan are often contradictory. On one hand, we're told to prepare for an invasion of Japanese robots, on the other, that the Japanese have done little more than copy US technology.

To sort out the facts from the rumors, *Robotics Age* went to Tokyo in October, to the 1981 International Robot Exhibition. Japanese robot-makers showed us their wares. They discussed their current research with our editors, and told us some of their plans for the future.

## THE JIRA IQ Test

How advanced are Japanese robots? How smart? The Japanese Industrial Robot Association (JIRA) has a classification scheme that can help answer these questions. JIRA classifies robots according to the way they receive control information. At the lowest level, for example, are

*manually controlled* arms. These telemanipulators are not considered robots at all in the US, and some experts accuse JIRA of inflating their robot counts by including them. JIRA, however, lists only six models of such machines in their 1982 directory.

On the next level are *sequence* machines. These robots are controlled by preprogrammed instructions. In *fixed*

sequence robots, the instructions are hard to change; in *variable* sequence machines, they are easy to revise. There are currently more sequence robots in operation than any other type—about 70%. But this percentage has been steadily falling as more sophisticated robots reach the marketplace.

*Playback* machines are teachable robots. A human moves the arm through a set of motions. The motions are memorized and, on command, can be endlessly repeated. Well-suited for simple tool and material handling, playback robots are becoming increasingly popular. JIRA's 1982 directory lists seventy-eight different models of them.

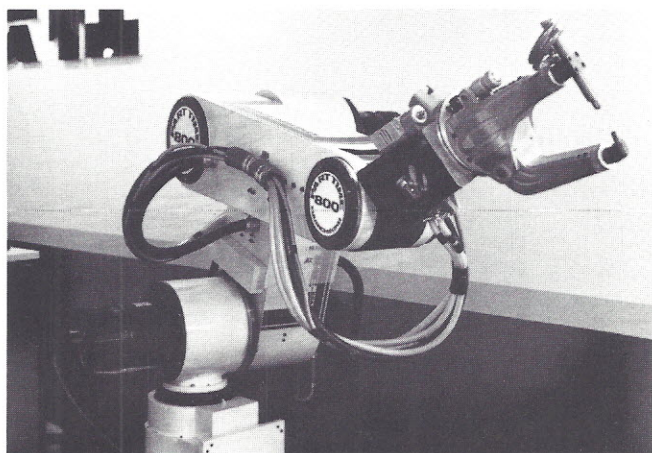
With *numerically controlled* robots, we enter the realm of external programming. These machines accept digitally encoded programs and, as the price of computing power falls, are likely to replace sequence machines in many applications.

Finally, at the summit of JIRA's classification scheme, is the *intelligent robot*. These—which purists consider the only "true" robots—can modify their behavior according to information they sense from the environment. At the lowest level, they have just a single sensor and the ability to process its signal. At the highest level, they represent the state-of-the-art in artificial intelligence research.



Over 120,000 people attended the 1981 International Industrial Robot Exhibition, held October 8-11 in Tokyo, Japan. Most were not roboticists, but local residents who share their country's fascination with high technology.

# 産業用ロボットの現状と展望



The show was dominated by spot welding, spray painting, arc welding, and pick-and-place robots. Here, Dainichi Kiko shows their new Part Time 800 robot, equipped with a spot welding end effector.

## Incremental Advance

Industrial robot shows always both impress and disappoint me. On one hand, it's almost hypnotic to watch the machines cycle through their precise movements: from the crane-like swoops of the Marol arm, which can swing 1000 pounds in its grip, to the nervous, cricket-like motion of the Picmat small assembly robot.

On the other hand, I am disappointed by the scarcity of intelligent robots. Like most robot shows in the US, this one was dominated by spot welders, spray-painters, arc welders, and pick-and-place machines. I saw no startling breakthroughs in robot intelligence.

I did see incremental advances. I saw design changes that make robots easier to apply, more versatile, and less expensive. Kawasaki Heavy Industries provides a good example. Although the original design of their PUMA arm came from Unimation, Kawasaki made important changes. "For our new painting robot," a company spokesman said, "we modified Unimation's design so the robot would fit into existing painting booths. The design allows the arm to retract close to the wall." The new design also substitutes electric for hydraulic drives on all axes except the base. This eliminates the danger of contaminating paint with hydraulic fluid.

At almost every company's booth, I could find some recent improvement in robot design. For instance: Hitachi's arc welding robot, Mr. Aros, has a comprehensive fault-detection system (detecting torch contact, arc failure, CPU failure, sensor failure, lack of welding wire, and pneumatic

failure); Fujitsu Fanuc uses bubble memory to store control programs, rather than floppy discs; and Yaskawa claims that their new Motoman L3 is the fastest robot on the market.

Japanese robots have become less expensive. Previously, the least expensive robot on the Japanese market sold for about ¥15 million (\$67,000). Now, Sankyo Seiko sells their new compact assembly robot for as low as ¥3.6 million (\$16,000). And Okamoto Seisakusho sells a ¥3 million (\$13,400) robot for loading metal sheets on press machines.

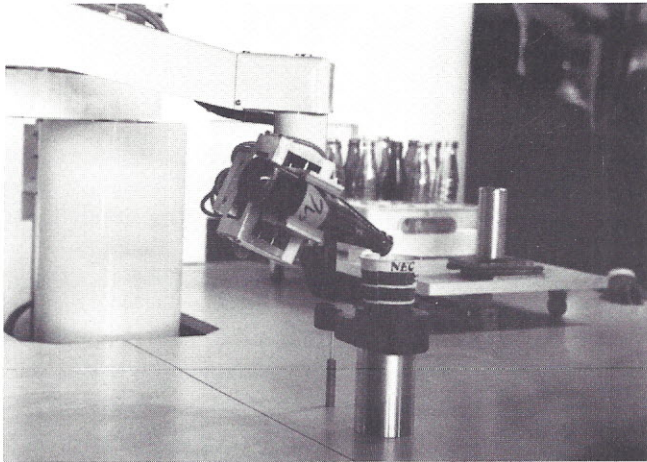


Assembly is the up-and-coming robot application. Here, Nitto displays their Picmat SCARA robot, designed for small parts assembly. SCARA stands for Selective Compliance Assembly Robot.

## Progress

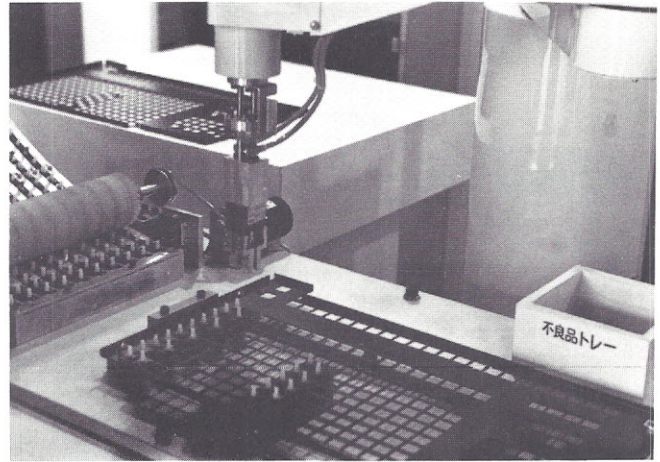
Japanese progress comes in waves. All of a sudden, a dozen companies fiercely enter the market with a new technology. As Makoto Kikuchi of Sony explained, "In Europe, if someone is engaged in work A, he feels that he must take up work B at all costs—because otherwise he cannot maintain his identity. In Japan, everyone will take up work A because of the strong inclination to join other people in work in the most current area."<sup>1</sup>

The trend now is to develop sophisticated sensors. Hitachi has done extensive work on a non-contact magnetic sensor for arc welding. National Pana-Robo, also building arc welders, has developed temperature, gas, magnetic, and pressure sensors. They claim to have a welding system that uses vision to track a weld seam.



*Nippon Electric Company (NEC) demonstrates their Assembly Robot for Precision Model-B. The robot pulls a bottled softdrink from a rack, pops off the cap, and pours it into a cup for some thirsty onlooker.*

The Japanese, like the Americans, are intensely developing vision systems. Mitsubishi and Kawasaki (with Unimation) both presented new systems at this year's International Symposium on Industrial Robotics (ISIR).



*With different tooling, NEC's Model-B does precision assembly.*

Both systems project collimated light beams onto the workpiece. They sense irregularities in the weld line through distortions in the reflected beam.

At the exhibition, Fuji Electric showed its CCD-based vision system. Used for automatic inspection, the Fuji system has done everything from checking medicinal tablets for defects to sorting vegetables by size and shape.

## LOOK MOM, NO WIRES!

FOR ROBOTICS  
COMPUTER HOBBYISTS  
REMOTE DATA ACQUISITION  
CONTROL PROJECTS  
TOYS  
AND MANY-MANY MORE CREATIVE APPLICATIONS

- USABLE WITH ANY COMPUTER ■ SERIAL INTERFACE ( RS232 )
- SMALL SIZE, LOW POWER ■ SINGLE SUPPLY ( 9 VOLTS )
- COMPLETELY DOCUMENTED ■ EASY TO BUILD ■ APPLICATION NOTES



The **DIGI-LINK** is a digital RF transmitter and receiver that links your robots, toys, appliances, remote stations or peripherals to your computer's intelligence and software based flexibility without any wires, cables, physical attachments.

Receiver and Transmitter.....	\$ 34.95
Receiver (w/ UART) and Transmitter.....	\$ 49.95
Receiver only.....	\$ 19.95
Receiver (w/ UART) only.....	\$ 35.95
Transmitter only.....	\$ 18.95

MAIL YOUR ORDER NOW! ENJOY THE **DIGI-LINK!!!**

Kits contain PCB, components, hardware, antenna, instruction and applications manual.  
CA residents add 6.5% tax.  
CHECK OR MONEY ORDER ONLY.

**E/Z ASSOCIATES**  
5589 Starcrest Dr.  
San Jose, CA 95123  
#408-578-8096

CIRCLE 18

## SYM-1 Single-Board Control Board

SYM-1 is the *first choice* for an on-board robotic control computer.

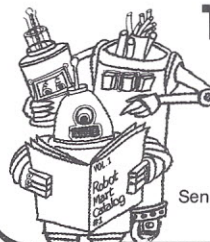
- Powerful 6502 microprocessor
- Bright 6-digit hex LED display
- Audible response keyboard
- 4K byte ROM resident monitor
- 1K byte Static RAM, with sockets for immediate expansion to 4K total expansion to 64K.
- Audio cassette interface with remote control, in either 135 baud (KIM-1 compatible) or 5 speed 1500 baud.
- RS-232 or 20 ma TTY interface
- 15 bidirectional TTL lines for user applications

**Special Introductory Offer—\$219**

## THE ROBOT MART

19 West 34 Street,  
New York, NY 10001

212/695-5108



Send \$3 for 1 year subscription to Robotics Catalog.

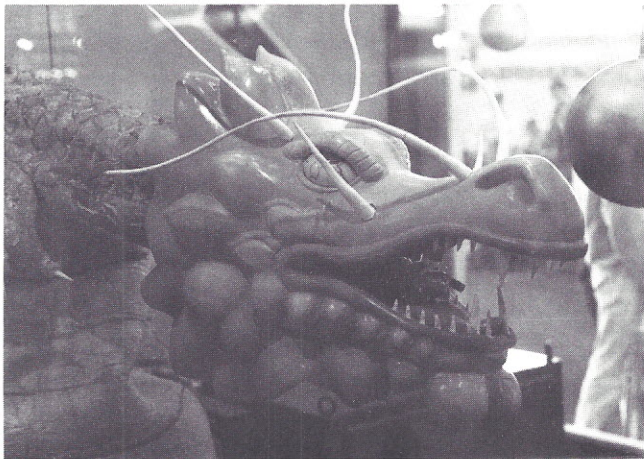
CIRCLE 24

Most of the robot-makers I spoke with—even those with no sensors on their current models—are now developing sensor systems. NEC is working on voice recognition, vision, force, and pressure sensors; Hitachi is exploring vision for assembly; Mitsubishi is researching vision; Nachi has vision and tactile sensing in the works—and I have named just a few. “In a few years,” a Nachi spokesman said, “Japan will compete in sensors.”

Assembly is the next major application the Japanese will tackle. As Kanji Yonemoto explains, “Automation of mass-production systems was the main current of industry up until the 1970’s; however, the predominant trend in and after the 1980’s will be the automation of small-batch production.” And this, he adds, will only be possible “as the technological development of intelligent robots makes progress.”<sup>2</sup>

Hitachi is making that progress. Their intelligent assembly robot combines two six-degree of freedom arms and a binary vision system under an heirarchical robot control language. Though not yet marketed, the Hitachi robot is as advanced as any assembly system used in the US.

And so, Japan may soon dominate advanced robotics as they now dominate consumer electronics. “We have no natural resources in Japan,” one company spokesman said. Then he smiled. “We do have technology.” □



Here is a robot dressed to kill. A Kobelco-Trallfa spray painting arm lurks behind that fearful exterior. The robot is jointly offered by Trallfa (of Norway) and Kobe Steel (of Japan). It's a good example of technology originating elsewhere, and improved by the Japanese.

# ROBOTICS

McDonnell Douglas Automation Company (MCAUTO), a divisional company of McDonnell Douglas Corporation, which ranks among the top three computer using corporations in the world, have excellent opportunities for talented professionals to do *Robotics-related work*. These positions provide an opportunity for growth, challenge and excellent benefits.

Responsibilities are to:

- Research new areas of Robotics technology and to explore potential areas of Robotics software development.
- Design, develop and implement Robotics software or mainframe, mini and micro computers.
- Apply advanced graphics techniques in Robot simulation systems.
- Develop and use advanced tools and techniques in pattern recognition computer-vision and artificial intelligence for applications in industrial automation and Robotics.

Varied levels of positions are available to individuals who meet the following qualifications:

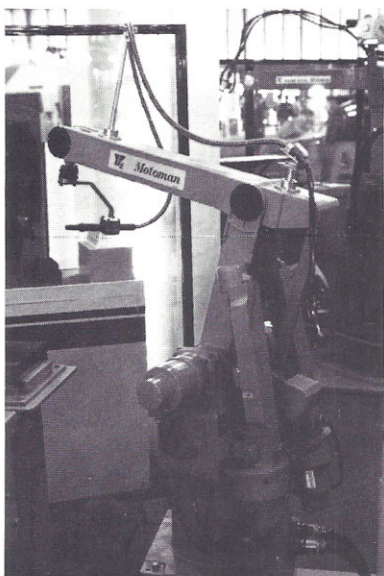
- Researchers
  - PhD in a technical field with strong computer sciences background.
  - Knowledge/experience in artificial intelligence required.
  - Knowledge/experience in computer graphics, N/C helpful.
- Programmer/Analyst
  - BS, MS or PhD in a technical field with strong computer science background.
  - Knowledge/experience in simulation and in Data Bases required.
  - Knowledge/experience in artificial intelligence, computer graphics, N/C, IBM and DEC systems helpful.
- U.S. Citizen or U.S. permanent resident required.

Please send your resume and salary history in confidence to:

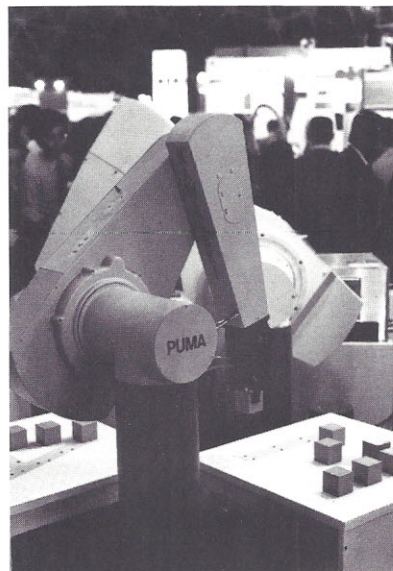
Manager, Professional Recruiting  
 McDonnell Douglas Automation Company  
 Department 62-266  
 P.O. Box 516  
 St. Louis, Missouri 63166

**MCAUTO**  
**MCDONNELL DOUGLAS**

An equal opportunity employer.



Another example of technological exchange is the familiar PUMA robot, born at Unimation, but now raised by both Unimation and Kawasaki.



Yaskawa builds this all-electric arc welding robot, the Motoman-L10. They claim that the sister model, L3, is the fastest in the world.

### A Note on the History of Japanese Robotics

It all seemed to happen at once. About twelve or thirteen years ago, Hitachi, Shin Meiwa, Mitsubishi, and Nachi began developing robots. Within a few years, Sankyo, Taiyo, Marol, and Mizano followed. They were all manufacturing companies; some, like Mitsubishi, were corporate giants. They began by perfecting their machines for in-house use, on their own welding and painting lines.

Then, four or five years ago, they began selling machines to each other and to other manufacturers in Japan. The production of robots grew—from ¥26.5 billion (\$118 million) in 1974 to ¥84.7 billion (\$378 million) in 1978.

1980 was a watershed year. Robot production grew 80% in value over the previous year, adding 19,000 new machines. The Japan Industrial Robot Association (JIRA) declared it "Robot Diffusion Year One."

In 1980, twenty-four robot makers and ten insurance companies formed the Japan Robot Leasing Company (JAROL). Robots could now be rented by small and medium-sized manufacturers. After one year of operation, JAROL had already rented \$5.5 million worth of robots and planned to rent \$12.5 million by the end of 1982.

Also in 1980, the Japanese government created incentives to spread the use of robots. They began giving special tax breaks to companies purchasing high performance computer-controlled robots. These companies can now take 13% of the purchase price as additional depreciation from April, 1980 to April, 1983. Japan's Small Business Finance Corporation and People's Finance Corporation began giving special financing to small and medium-sized firms who improve worker safety by purchasing robots.

Robot exports—which most companies have previously not pursued—are now a high priority. Within five years, exports should rise from 3% to 16% of Japan's robot output.

Will they market in the US? *Robotics Age* asked this question to robot-makers at the exhibition. NEC answered yes, from now on; Shin-Meiwa said they were just starting; Hitachi already has a US office in New Jersey; Mitsubishi plans to start soon; Fujitsu Fanuc already has a subsidiary in the US (General Numeric, Chicago, Illinois); Nachi is already selling five models in the US; Sankyo has established US sales; and Taiyo is not yet selling in the US—but they say they hope to do so soon.



## For the first time, a really practical vision system for automatic on-line industrial inspection and q.c.



The author (left) shakes hand with a Matsushita robot designed to demonstrate its sensors. "Wow, what a grip!" it said.

Field proven ORS vision systems can now provide production and quality control engineers with an off-the-shelf solution to many types of on-line inspection applications. Automated high speed visual sorting, classifying, routing, defect identification, character recognition and robot vision applications are now feasible with a cost effective ORS system. Microcomputer-based and available with a range of sensors, real-time ORS systems can accept both incremental and continuous motion (up to five or more objects per second) and are ruggedized to operate in most industrial environments. Visual decisions are based on the overall information content of an object including grey scale data, spatial relationships and shape. A variety of output signals are provided as standard.

For more information, call or write:

Marketing Director  
Object Recognition Systems, Inc.  
521 Fifth Avenue,  
New York, N.Y., 10017  
(212) 682-3535 or (609) 924-1667.

CIRCLE 18

### US Experts on Japanese Robots

"The Japanese do not have a technological edge at all. They only have a fantastic ability to take an idea and make it work."

—Joseph Engelberger, founder and president of Unimation Inc.

"There is no big, advanced robot system in Japan that even equals the one we created by combining the PUMA with the MIC vision system...The Japanese have succeeded mostly in proliferating established machines."

—Dr. Charles Rosen, founder and chief scientist of Machine Vision Corporation.

"Japan's success in robot production and installation can be traced, in large measure, to its labor practices. The Japanese employee in a major corporation is guaranteed lifetime employment (until the age of 55-60). In addition, all

employees receive two bonuses, each ranging from 2-5 months pay, in June and December, which are ultimately based on the company's profitability. The worker, not fearing loss of a job, does not oppose automation; in addition, as automated production generally enhances profit and quality and consequently the bonus, Japanese employees welcome the robots."

—Paul Aron, Daiwa Securities America Inc., *Paul Aron Reports* (#22): *Robots in Japan*, 1980.

"The Japanese don't have the same problem with stockholders and the bottom line that we do in this country. In Japan, the banks finance companies, not the general public who finance American firms.

—George Devol, founder and president of Devol Research.

## STEPPER MOTOR/DRIVER BOARD

Interface Your Computer to the REAL World  
Indispensable for Robotics — Many Control Applications

### FEATURES

#### Stepper Motor 101-SM

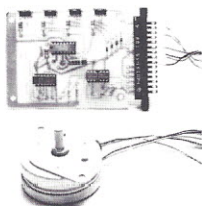
(N.A. Philips K82701-P2)  
Operating Voltage 12 VDC  
Weight 8 oz  
Holding Torque 10.5 oz-in  
Rotation 7.5° per step  
Max. Running Rate 850 steps/sec  
Step Angle Tolerance ±0.5° (non-cumulative)

#### Driver Board 2003-DB

Dimensions 4.5"×3.2"×0.83"  
Power Requirements 9.5-18 VDC  
Digital Inputs 4-20 VDC  
Max. Current per winding 2 amps  
Driver Chip SAA 1027  
Optically isolated

#### Linear Actuator Stepper Motor

**501-AM** (N.A. Philips L92121-P2)  
Operating Voltage 12 VDC  
Weight 1.5 oz  
Max. Force Exerted (energized) 21 oz  
Min. Holding Force (unenergized) 40 oz  
Travel 0.002" per step  
Max. travel 1.88"  
Max. Pull-in Rate 425 steps/sec  
Max. Pull-out Rate 650 steps/sec



### PRICES

**101-SM** \$28.95  
**501-AM** \$41.95  
**2003-DB** \$44.95 (includes edge connector and heat sinks)

Optional on-board **Oscillator** and 20-turn **Potentiometer**  
(add \$6.00 and add suffix "-OP" to 2003-DB)

**SAA 1027** \$16.50

Other Stepper Motors: inquire about availability.

Terms: Check or money order U.S. funds only. Add 3% for shipping and handling. (N.Y. state residents add 7% sales tax.) Immediate delivery from stock. Data and schematics are included. All items guaranteed.



P.O. Box 56  
Bellport, New York 11713  
(516) 286-2144

CIRCLE 19

### Notes

- [1] Gregory, Gene Adrian and Etoni, Ahio, "Japanese Technology Today," *Scientific American*, October 1981, p.J8.
- [2] Yonemoto, Kanji, "The Socio-Economic Impact of Industrial Robots in Japan," *Proceedings of the 11th International Symposium on Industrial Robots*, 1981, pp.2-3.

### Further Reading

Aron, Paul. *Paul Aron Reports (#22): Robotics in Japan*. New York: Daiwa Securities America, Inc., 1980. (Their address is One Liberty Plaza, New York, New York, 10006.)

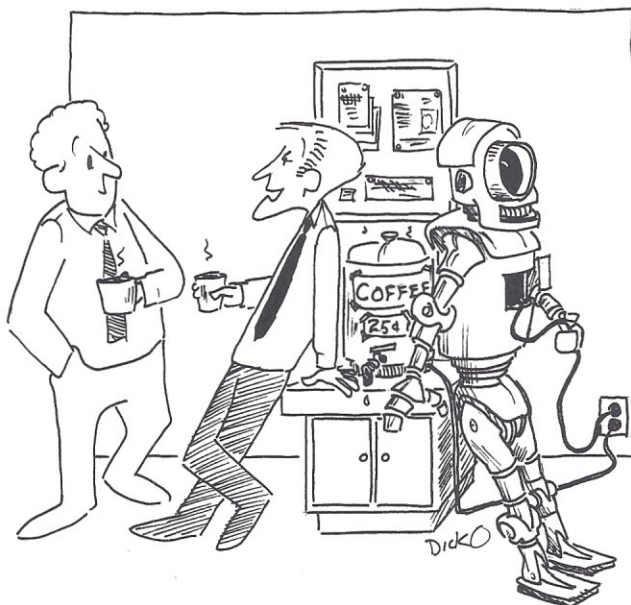
Gregory, Gene Adrian and Etori, Akio. "Japanese Technology Today," *Scientific American*. Vol. 245, No. 4, October 1981. (Although this is actually a 46 page advertisement, it is a good source of information on the Japanese electronics industry.)

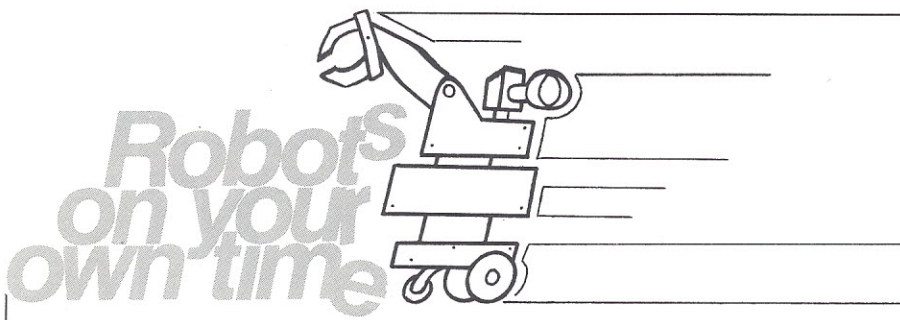
Kohno, Michinaga, Hiroshi Horino and Mitsunobu Isobe. "Intelligent Assembly Robot," *Hitachi Review*, Vol. 30, No. 4, (1981).

*Proceedings of the 11th International Symposium on Industrial Robots*, Tokyo: Society of Biomechanisms Japan (SOBIM) and Japan Industrial Robot Association (JIRA), 1981. (You can get a copy from JIRA, 3-5-8 Shiba Koen, Minato-Ku, Tokyo, Japan. The price, including seamount, is ¥21,8000—about \$95.)

*Robotics Industry Directory, 1982*, Tujunga, California: Robotics Publishing Corp., to be published in January, 1982. (For more information, see the advertisement on page 6 of this issue.)

*The Specifications and Applications of Industrial Robots in Japan, 1982*, Tokyo: Japan Industrial Robot Association (JIRA), 1981.





by  
John Blankenship  
Devry Institute of Technology  
Atlanta, Georgia

# TIMEL:

## A Homebuilt Robot

### PART 2

In the July/August issue of *Robotics Age*, I discussed the mechanical construction of TIMEL (Truly Intelligent Mechanical-Electrical Life). One of my major points in that article was that the real challenge in creating an intelligent home robot lies in the software that controls the robot's behavior. My design of TIMEL (Photo 1) was intended to provide the cheapest hardware to serve as a testbed for developing intelligent software.

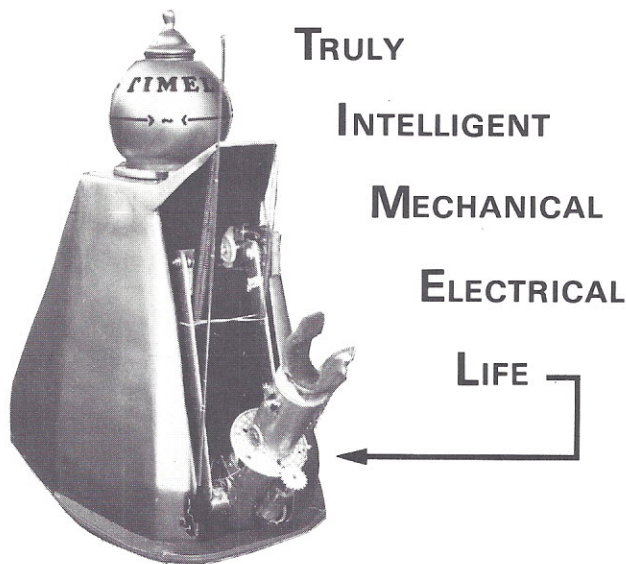


Photo 1. TIMEL: Truly Intelligent Mechanical-Electrical Life? The simple robot pictured here is one example of a system that will enable hobbyists to experiment with the software that make intelligent robots a reality.

This article will continue with the details of TIMEL'S construction—this time covering its internal electronics and its radio link to an external computer. Although I expect many of you to build TIMELs of your own, I don't expect you to build a duplicate of mine. Again, the idea of TIMEL is to use the simplest components available to you and to assemble them to suit your own interests.

Because of these expected variations (which I encourage) I'll be taking a somewhat different approach in explaining TIMEL's electronics. Instead of making this a step-by-step construction project, I'll take a more tutorial approach, designed to give you an understanding of the circuits I used so that you can adapt them to your own TIMEL.

#### Basic Design Requirements

I had several objectives in mind when I designed the control system for TIMEL. These were: an ability for direct manual control, control from an external computer, control from an on-board computer, communication between the two computers, and a failsafe, hard-wired "nervous system" to protect the robot.

Manual control is important for several reasons. One is convenience—there are many occasions when you want to move the robot without having to write a program or to carry it. Manual control is important for debugging the hardware, and it can give you personal satisfaction to see the robot move during the course of a long construction project. Even after the system is complete, manual control gives important clues to writing intelligent programs—if you can't do a task manually, how can you program it?

Most computers with a high-level language and enough compute power for sophisticated robot control programs are too bulky to fit inside a simple frame like TIMEL's—especially if you consider the keyboard, disks, monitors, etc. that are essential for program development. Battery power of such systems is also a problem. I decided to use an external APPLE II computer as TIMEL's controller and have it communicate with the robot on a radio link. The link can be adapted to any computer with a parallel output port.

The bandwidth of an inexpensive radio link (especially mine!) is usually too low for many of the real-time control functions the robot needs—monitoring sense switches, ultrasonic range measurement, counting wheel revolutions, etc. Using a small single board computer (SBC) on board solves this problem. The KIM I used can perform all these functions while still monitoring requests from the APPLE. The KIM can override external commands if necessary to protect the robot. This hard-wired “nervous system” provides failsafe protection—just in case software on the APPLE is faulty.

### A Simple Radio Link

There are three ways of establishing a radio link from the external computer: buy a commercial unit that does just what you want, adapt some other type of unit, or build your own from scratch. In keeping with my design philosophy for TIMEL, I picked the method that required the least amount of effort and expense in hardware. Using a standard model airplane remote control (RC) unit provides the simplest manual control method and is easy to interface to the external computer. Photo 2 shows the RC transmitter (and also my KIM computer).

Rather than decode the signals from the on-board RC receiver for a direct electronic interface, I decided, again taking the simplest route, to use the servos that came with the RC unit to trip switches that control the robot. This approach leaves the RC receiver unmodified and eliminates any decoding circuitry.

Before you start to write letters cursing the inefficiency of such a system, consider the following: TIMEL is not meant to be state-of-the-art hardware, just a testbed for software. Depending on the brand of RC unit you use, the decoding may either be in the receiver, in the servo modules, or (most likely) distributed between them. To build your own decoder would require a custom design and additional hardware. Considering that the servos usually come with the RC unit, the mechanical connection

# COMING SOON

- The AL Robot Control Language
- Introducing the Rhino XR-1
- Understanding Natural Language
- Advanced Vision Hardware
- The Future of Personal Robotics
- The Winners of the Home Robot Photo Contest

*and MORE!!!*

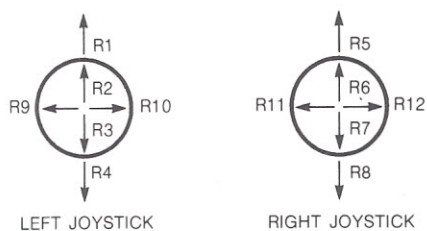
## READER SERVICE

PRODUCT #	PRODUCT
1	CEERIS INTERNATIONAL, Assembly Robot, p.47
2	IVS, INC., Realtime Image Processing System, p.47
3	GENERAL INSTRUMENT MICROELECTRONICS, Speech Synthesis Support, p.47
4	ROBOTIC SYNERGY, Stepper Motor Controller, p.47
5	CYBERNETIC MICRO SYSTEMS, Stepper Control Kit, p.48
6	SYSTEMS INNOVATIONS, Micro for Control Applications, p.48
7	DATRICON CORP., Applications Controller, p.48
8	ILC DATA DEVICE CORP., A/D Converter, p.48
9	ILC DATA DEVICE CORP., 12 Bit A/D Converter, p.49
10	ILC DATA DEVICE CORP., 14 Bit S/D & R/D Converter, p.49
11	DATTEL INTERSIL, D/A Board, p.49
12	SME, Industrial Robot Books, p.21
13	TEKSYM, Calculating Process Monitor, p.50
14	SCOTT ENGINEERING, Rotating Swivel for Hot Glue, p.50
15	HOBBY ROBOTICS, Robot Kits, p.45
16	GENERAL INSTRUMENT MICROELECTRONICS, 1982 Product Catalog, p.50
17	STOCK DRIVE PRODUCTS, Components Handbook, p.50
18	E/Z ASSOCIATES, Digi-Link Radio Link, p.32
19	AMSI CORP., Stepper Motor Driver Board, p.36
20	ORS, Vision System, p.35
21	SANDHU MACHINE DESIGN, INC., Rhino XR1 Robot, Inside front cover
22	ADVANCED COMPUTER PRODUCTS, Mini-Mover 5 Robot, p. 15
23	NORDSON CORP., Nordson Robots, pp.2,3
24	ROBOT MART, SYM-1 Single Board Chip, p.32
25	DATASOFT, LISP for Microcomputer, p.8
26	ROBOTICS INDUSTRY DIRECTORY, 1982 Directory, p.6
27	SORRENTO VALLEY ASSOCIATES, Robotics Development, p.4
28	DORING ASSOCIATES, Mini-Mover 5 Robot, p.28
29	STONERIDGE TECHNICAL SERVICES, Speech Newsletter, p.16
30	MCDONNELL DOUGLAS, Employment Opportunities, p.33
31	UNIV. OF RHODE ISLAND, Robotics Research Careers, p.53

## READERS ADDRESS THE EDITOR

This is your channel to the Editor. More articles will be included in future issues by authors and on subjects which receive high ratings. Mark your ratings on the facing card.

Article	Title
A	Teach Your Robot to Speak
B	Fast Trig Functions for Robot Control
C	An Interview with George Devol
D	The Great Japanese Robot Show
E	TIMEL: A Homebuilt Robot, Part II



	HORIZONTAL POSITION	MODE	VERTICAL POSITION	
			L. JOYSTICK	R. JOYSTICK
L. JOYSTICK	LEFT	R9: ARM	SHOULDER	ELBOW
	RIGHT	R10	COMMUNICATION MODE	
R. JOYSTICK	LEFT	R11: WHEELS	L. WHEEL	R. WHEEL
	RIGHT	R12: HAND	WRIST	FINGERS

Figure 1. The transmitter joysticks can activate twelve different switches, according to the chart shown above. The table shows which functions these switches control.

is the most economical, and it is entirely adequate for simple control of all the motors on the robot. A complete RC system costs about \$200 new, and you can save a lot by checking with your local model airplane club for a used unit.\*

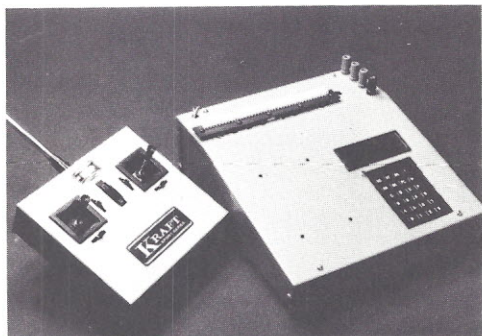


Photo 2. A model airplane radio control unit transmits commands to TIMEL, either by manual joystick movement or external computer control.

My unit has four channels controlled by the horizontal and vertical movements of two joysticks. Since I needed to control several different functions (hand, arm, and wheels), I decided to use the horizontal positions of the joysticks as a function selector and their vertical positions to control the rate and direction of movement. Figure 1 shows the mapping between the joystick positions and the receiver switches (R1-R12) that control the various functions.

*\*Editor's note: The availability of inexpensive computer-to-computer radio links may make an all electronic interface to your robot more attractive. (See, for example, the Digi-Link advertised on page 39.)*

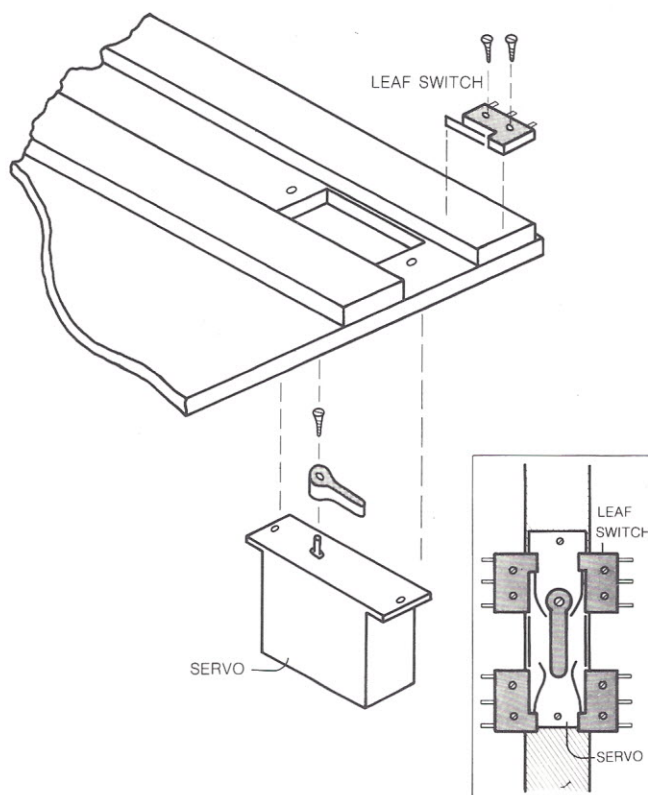


Figure 2. The mechanical action of the servos can cause small leaf switches to close. These switches are connected to the control electronics to activate TIMEL's motors or send signals to the onboard computer.

Vertical movement of each joystick controls four switches. For example, if you move the right stick forward, R6 closes. If you move it further forward, R5 also closes. Thus, R5 and R6 correspond respectively to the slow forward and fast forward movement of the function(s) selected by the left/right positions of the joysticks.

As another example, consider moving the right stick left, which selects the wheel functions. Vertical movement of the left and right sticks control the left and right wheels, respectively. Moving the left stick forward moves the left wheel forward (backward moves it in reverse). Moving the right stick controls causes the wheel to move more quickly. Moving the right stick controls the right wheel similarly. But you have to keep holding it to the left; this keeps the wheel functions selected.

Figure 2 shows how I mounted the servos to control the switch closures. In response to pulses sent by the RC transmitter and decoded by the receiver, the servo shaft rotates the lever arm left or right. The amount of movement is proportional to the joystick's movement. As the lever rotates, it trips one or more leaf switches. On the two vertical servos, I mounted four switches with the leaves bent so that one switch would activate before the other.

## TIMEL's "Nervous System"

The receiver switches R1-R12 are only part of the inputs to the nervous system. There are also fifteen sense switches, S1-S15, that monitor various conditions. For the discussion that follows, refer to Figure 3, which shows where these switches are located.

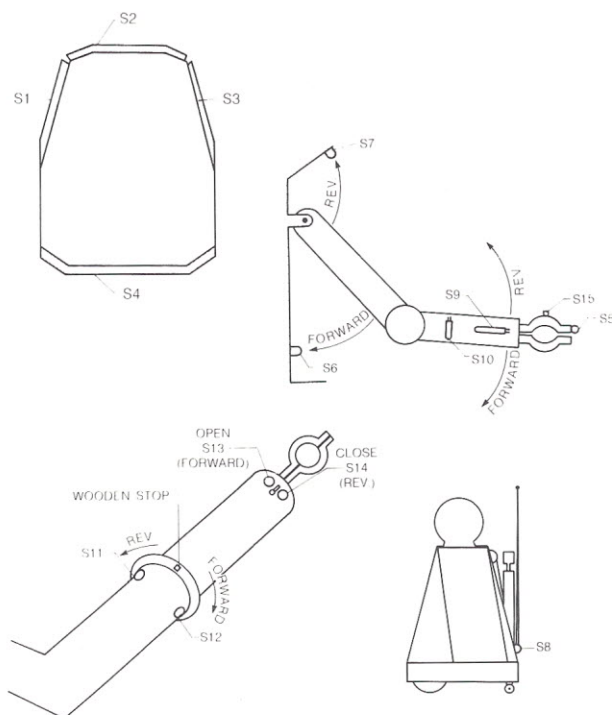


Figure 3. The diagrams above show the locations of fifteen sense switches. Each switch inhibits the motion of one of TIMEL's motors in the event of a collision.

The first four switches are made from pieces of ribbon switch. These are bonded with silicon glue to foam rubber around the base of the robot. They provide immediate shutdown of the drive motors in the event of a collision. (Eventually, ultrasonic ranging, monitored by the KIM, should prevent collisions in advance.) Photo 3 shows a close-up of a bumper switch.

To be most effective, the nervous system should have some built-in intelligence. For example, contact on S4 should void any attempt to move either wheel in reverse and all motors, including the wheels, to move forward. Similarly, contact on S1 should inhibit reverse motion of the left wheel and forward motion of the right.

To detect a collision with an overhanging obstacle, S8 is mounted at the base of the antenna (see photo 3). S5, on the end of one finger, works the same as S2; both can detect a collision while moving forward.

If the arm reaches its limits, S6 and S7 shut down forward and reverse motion of the shoulder. But using absolute limits on the elbow joint is not sufficient for safety. Instead, I mounted mercury switches S9 and S10 to keep the forearm from ever going beyond the vertical, either at the top or the bottom of its arc. To limit wrist movement, S11 and S12 are tripped by a small block glued to the rotating wrist gear. Similarly, S13 and S14 are tripped by a screw on the finger cam when the cam moves to its extreme positions.

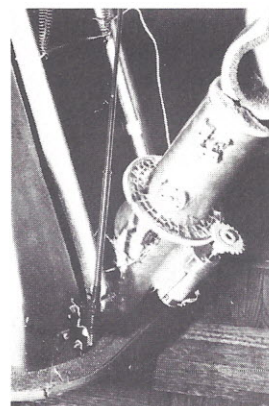


Photo 3. Numerous sense switches serve as inputs to TIMEL's "nervous system."

One final switch, S15, is mounted on the side of the hand to stop elbow movement in the event of a collision. To make best use of S15, TIMEL's hand should be turned with S15 either up or down, depending on the direction of elbow movement. I mounted the wrist motion limiters so that S11 and S12 will stop the hand with S15 either up or down. Should S15 close, only shoulder movement will move the arm to back off from the obstacle.

Receiver and sense switches are the inputs to TIMEL's nervous system. Of course, they are configured to suit the mechanical design of my robot. Should you choose a different design, you'll need a different switch configuration. In presenting the schematic of TIMEL's nervous system, I'll explain the design so that you can easily adapt the circuit to your own robot. I used only four types of logic gates in TIMEL's nervous system. The packages and their pin assignments are shown in Figure 4. All are standard TTL with +5V supplied by the circuit in Figure 5.

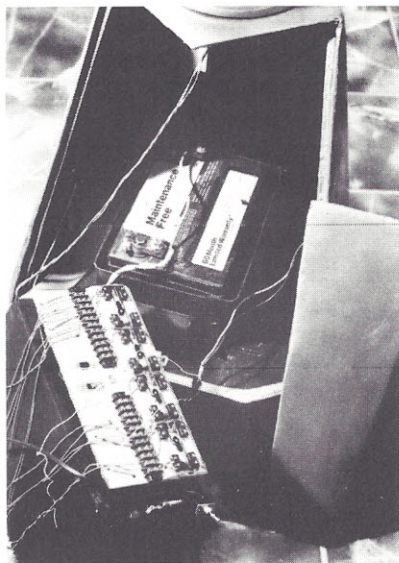


Photo 4. The servos from the RC unit, mounted next to leaf switches as shown, are the mechanical connection between TIMEL and the external world.

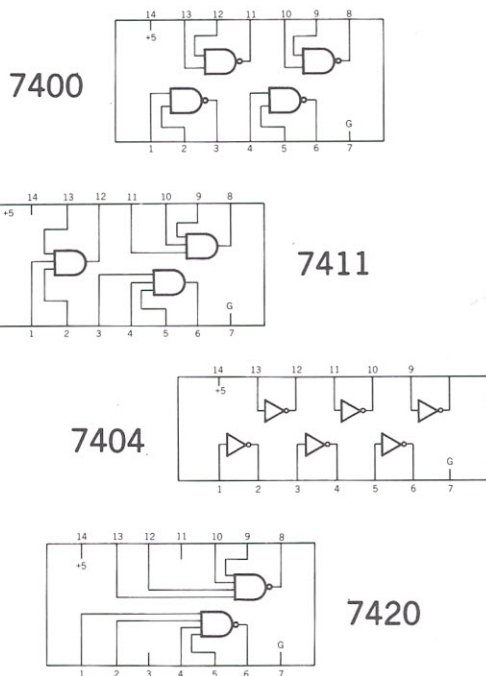


Figure 4. Logic packages used in TIMEL's control circuit. For low power consumption, I recommend using 74C—, 74L—, or 74LS— devices (in that order).

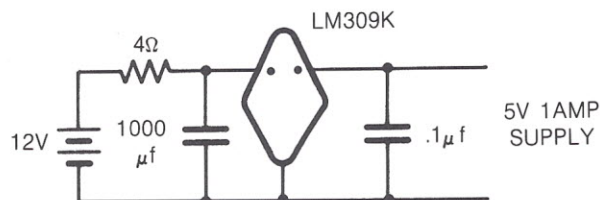


Figure 5. A simple 5V power supply that uses the onboard 12V battery as a power source.

Consider the “meaning” of a logic signal. Although the line may be either low or high (0 or 1), either state may be the one that indicates the “active” condition of that signal. In TIMEL, for example, all the lines from the sense and receiver switches are normally high. When a switch closes, it grounds the line to indicate that the condition it monitors has occurred. Thus, all these lines are “active low.” This is important when it comes to reading the schematic.

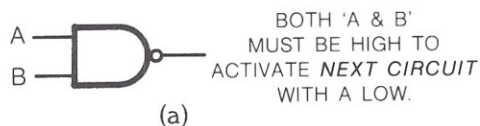


Figure 6. Drawing a gate according to the function it performs makes reading the schematic easier. For a 7400 NAND gate, (a) shows it performing an AND function, and (b) shows its use to perform an OR. In Figure 7, I draw each type of gate according to how I use it.

The way you draw a gate can make the schematic easier to read. The output of a NAND gate in a 7400 package can be considered as an active low AND of two active high inputs (Fig. 6b). The same can be said of the other multi-input gates I use. In the schematic for TIMEL's logic, shown in Figure 7, I give all the gates according to their intended function.

The schematic can be divided according to the functions performed by different sections. Transistors Q1-Q3 control the functions commanded by *vertical* movement of the left joystick. Q7-Q9 perform the same functions for the right joystick. When Q1 is turned on by a positive voltage at its input, it closes relay Y1, applying 12V to the diode chain. The four diodes drop about 2V, leaving 10V for “slow speed” output to the motors. If Q2 turns on, Y2 will bypass the diodes and pass 12V to the motors for “full speed.”

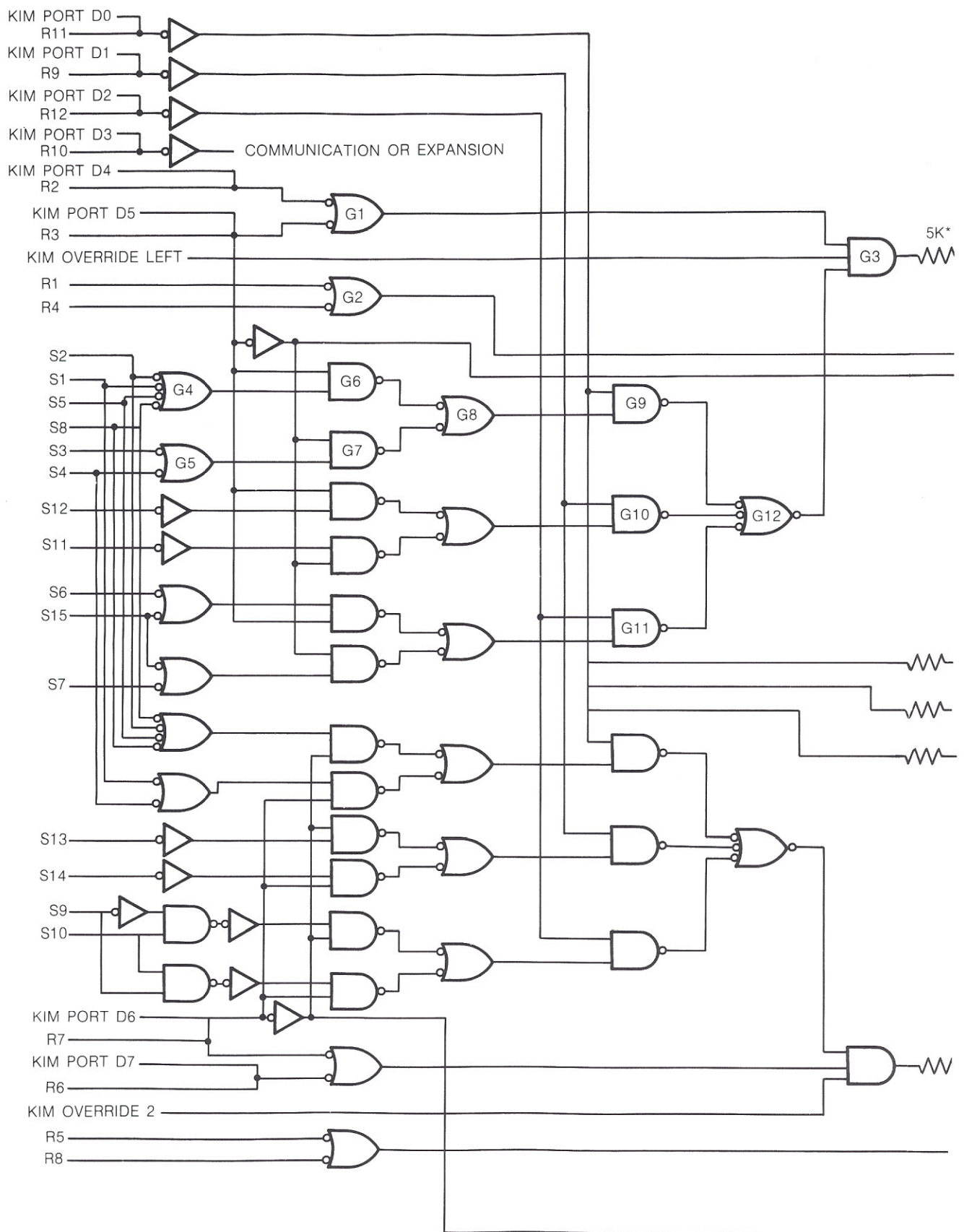
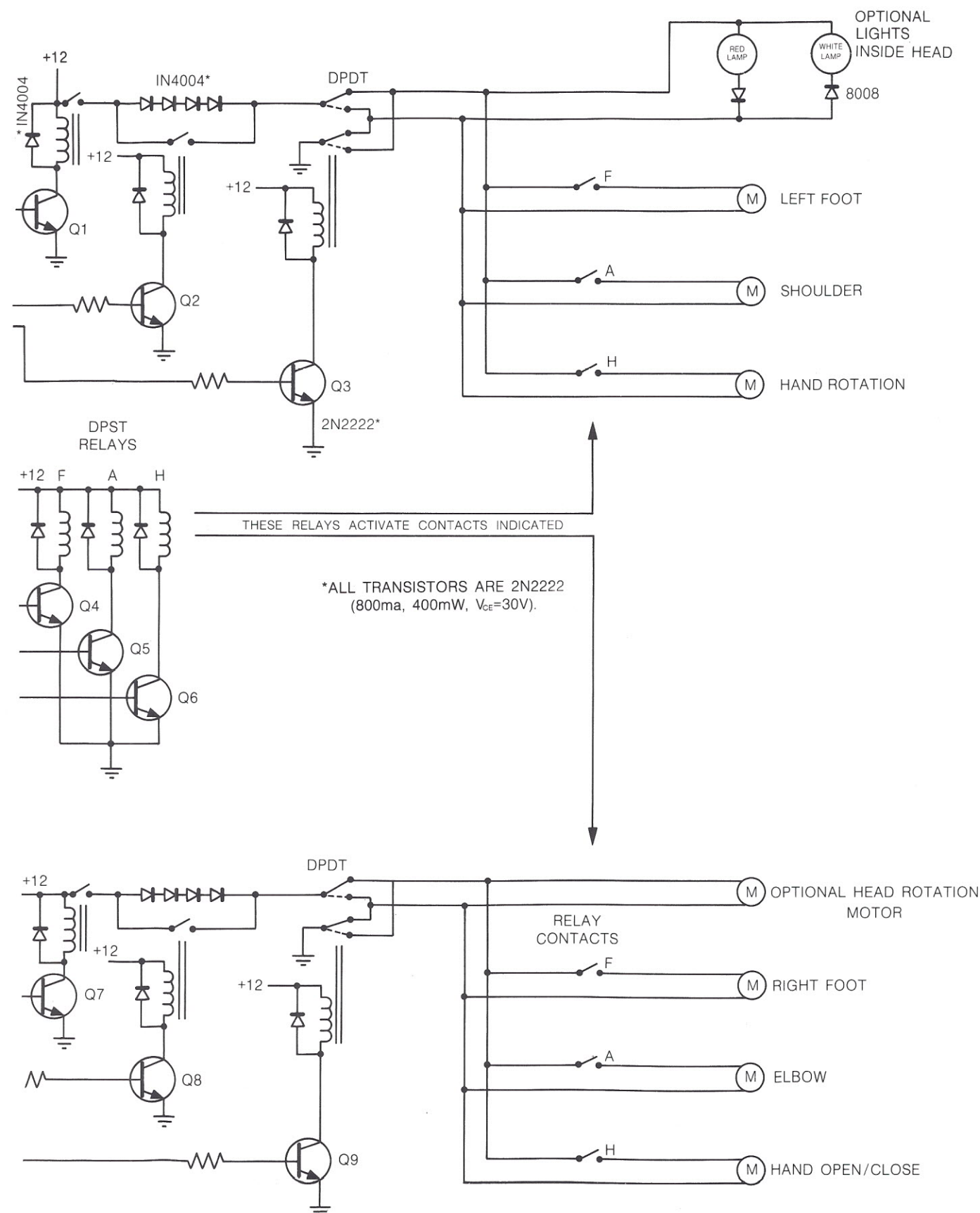


Figure 7. *TIMEL's nervous system interfaces to the internal and external computers and automatically overrides any command that presents danger to *TIMEL* or its environment.*



You'll need to experiment with the number of the diodes to give a satisfactory "slow speed" with the motors you use in your robot, and choose their current rating according to your motors' requirements.

Q3, when active, reverses the polarity of the output voltage. Choose relays with ratings sufficient to carry the motor's drive current throughout its operating range, and then pick switching transistors hefty enough to carry the relay's drive current. For small relays, almost any NPN transistor will do. For larger ones, you may need to use Darlington transistors. Start with a 5K resistor on the transistor's base and experiment for the best results.

Transistors Q4-Q6 and their associated relays are energized by the horizontal movement of the joystick to select one of the movement functions: arm, wheels, or hand. Even though vertical joystick motion has enabled output power, nothing will happen unless you also select a function, thereby passing the output voltage to the appropriate motor.

The next thing to consider is how to get a one (active high) on the inputs of the transistors. This is the function of the mass of gates on the left of the schematic. The inputs to this logic network come from the receiver switches, the sense switches, and the outputs of the internal KIM computer.

To use the computer interface for both monitoring and override control as I have done, you must use one whose lines can be programmed to be either inputs or outputs. Examples of such interfaces are the Programmable Interface Adaptor (PIA) and the Versatile Interface Adaptor (VIA). Using the PIA on the KIM, the control logic can reprogram them as outputs to override the external commands.

Now let's look at the hard-wired logic functions. I will explain how two of them work—that should make the others clear, too.

Transistor Q2 controls the speed as commanded by the left joystick. It should close the relay whenever the stick is moved all the way forward or backward. These conditions correspond to closing R1 and R4, respectively. Referring to the circuit, note that R1 and R4 connect to a NAND gate, G2. G2 is drawn as an OR function to help indicate how we are using it to take the OR of the two active low signals. If either R1 or R4 is zero, the G2 will output a high (1), closing the "fast" relay Y2.

The on/off action of the left stick, controlled by transistor Q1, is more complicated. In general, Q1 should be on whenever R2 or R3 are closed (low). The AND gate G3 at the base of Q1 needs all three inputs high to activate the relay. These inputs are the basis for understanding

how the "nervous system" works, so we'll look at each input in turn.

G1 takes the OR of the two receiver switch signals. If either switch closes, it outputs a one. If the other inputs to G3 are high, Y1 closes and gives power to whatever left-stick function you've selected. Note that a PIA I/O line is connected in parallel with each of the switches. Normally, these lines would be set as inputs so that the KIM could monitor the movement commands. If they were programmed as outputs, setting a line to zero would cause the same results as closing the corresponding receiver switch.

The middle input line to G3 is the KIM override line. It is normally programmed to hold a one, but the KIM can disable the action of receiver switches R2 and R3 by outputting a zero.

The bottom input to G3 is the automatic override from the nervous system. The 3-input AND gate G12 is drawn to indicate that if any of its inputs goes low, it will output a low and disable Q1. The gates that produce these inputs, G9-G11, are used to monitor certain groups of sense switches, depending on which movement function you've selected. G9, for example, can only override when the left wheel function is requested (R11 closed or D0 low). In this case, it inhibits movement if G8 outputs a 1.

The inputs to G8 are provided by G6 and G7, which are active only during forward or reverse motion, respectively. If you move the left stick forward, R2 will be low and R3 will be high, placing a one on one input to G6 and disabling G7 with a low. Closing R3 by backward stick movement reverses the situation. With R3 high any of the sense switches S1, S2, S5, or S8 inhibit movement when closed. G4 takes the (active high) OR of these four (active low) signals. Refer to Figure 3 to see why these switches override forward motion of the left wheel.

Similarly, S3 or S4 override reverse motion of the left wheel through G5 and G7. This type of logic is applied through the rest of the circuit. After studying a few more examples, you should be able to thoroughly understand its operation and, more importantly, be able to design a nervous system for your own robot.

## Remote Computer Control of TIMEL

For direct manual control of the robot, we can just move the joysticks on the RC transmitter. But to permit a stationary external computer to control it, we need a way to "move" the joysticks electronically.

Figure 8 shows a circuit that can accomplish just that. The 4016 ICs are CMOS analog switches, controlled from

an output port on my APPLE (any parallel port will do). I wired the switches in parallel with the joystick potentiometers in the transmitter. (The transmitter circuitry is not shown.) A pair of switches control one direction (horizontal, vertical) of one joystick. When one switch of a pair is turned on, it connects a 6.8K resistor and its associated trim pot in parallel with one side of the joystick pot. This reduces the resistance of that side and, with the joystick centered, makes the transmitter think you've moved the stick in the corresponding direction.

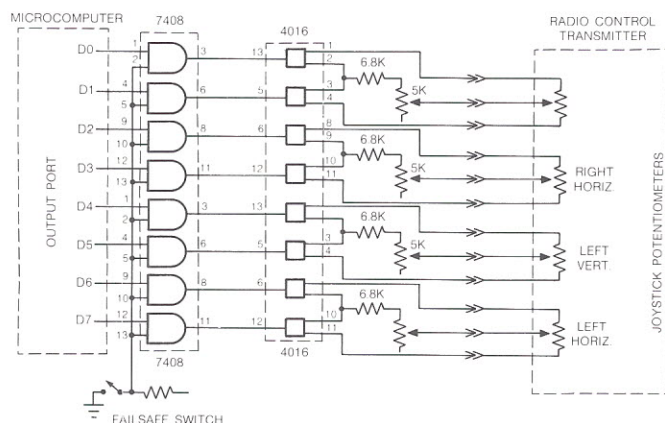


Figure 8. The 4016 analog switches allow any external computer with a parallel port to simulate movement of the joysticks. The resistance values will vary for different brands of RC units. The trim pots permit fine adjustment of how much the computer "moves" the joysticks.

With this circuit, the APPLE can only control TIMEL with one speed. Which speed that is depends on how much the resistance changes when the analog switch conducts. (Adjust the trim pot to get the speed you want—the values shown worked for my RC rig.) You can add additional switches to get two-speed control if you wish. One warning: be sure not to turn on *both* switches of a pair at once—you'll short out the transmitter pot and possibly do some damage.

### Using the System

Through the RC transmitter, either manual joystick motion or external computer commands can control each of TIMEL's motors. The internal computer can monitor or override the movement commands and perform whatever other sensory or control functions you choose to add. In the communications mode, the KIM can read the vertical

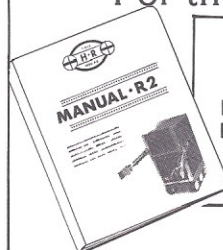
stick movements (whether manually or electronically originated), but no motors will be activated. You could use this mode to request the KIM to perform some pre-programmed maneuver.

Of course, the sense switches will automatically override any erroneous command, regardless of its source. In such a case, it's up to you or your software to analyze the situation and take appropriate corrective action.

In the future, I'll be writing about TIMEL's control programs in the KIM and the APPLE. However, as I've said before, I view software development as an evolutionary task that I'll never finish. New ideas come much faster than I can add them to the programs. But then, that's what I think personal robotics is all about. □

## HOBBY ROBOTICS' Shoppers Mart

— For the SERIOUS Hobbyist —

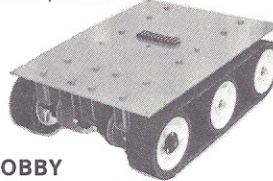


Hobby Robotics Directory

Amateur Robotics Designers Newsletter

ARD Newsletter. Quarterly. Keeps you abreast of latest news, perks up interest! \$6.00

**Complete RU-2 Manual!** Everything you need to build your own robot shown, detailed! Build from base up with Hobby Robotics' highly maneuverable 2-ft. per sec. super-traction base, extendable arms lift-rated at 15 lbs., grip-like hands. Manual only \$25.00, refundable with first purchase.



**HOBBY ROBOTICS' ROBOT BASE**  
Ready to roll!  
**ONLY \$495** (includes shipping)

**Hobby Robotics' Parts Supplier Directory.** Chock-full of names, addresses, phone numbers you want for hard-to-find parts, systems, build-your-own-robot info. Only \$3.50. Get yours now!

### HOBBY ROBOTICS

5070 Buford Highway  
Norcross, Ga. 30071 404/448-0190

Please send items checked below:

- ☐ Qrt. Newsletter ☐ RU-2 Manual  
☐ Parts Directory ☐ Robot Base  
☐ Hobby Robotics' Catalog ☐ FREE  
Total Enclosed \$\_\_\_\_\_ (includes shipping)  
☐ Personal Check ☐ Cashiers Check/M.O.  
☐ Mastercharge ☐ Bank Americard

Acct. # \_\_\_\_\_  
Signature \_\_\_\_\_ Exp. \_\_\_\_\_  
Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_

CIRCLE 15

# Eye to Industry

by Seiichiro Yahagi

All across the US, roboticists are closely watching the developments in Japan. A glance at the statistics will tell you why:

In 1980, the Japanese production of industrial robots rose to ¥78.4 billion (\$350 million), increasing 85% over the previous year. In terms of numbers, the year's output was 19,000 units, up 35% from 1979. According to the **Japan Industrial Robot Association (JIRA)**, robotics in Japan will emerge as a ¥100 billion (\$450 million) by the end of 1981. Since demand for industrial robots should grow at an annual rate of 50%, robotics could become a staggering ¥500 billion (\$2.2 billion) business by 1985.

By robot type, 1980 saw 33% more assembly robots built than 1979. Three times as many arc welding robots, twice as many painting robots, and 60% more press processing robots rolled off the assembly line in 1980 than in 1979. The electrical industry proved the best customer for these machines, accounting for 36% of the total number of shipments. The automotive industry, previously the number one robot-buyer, now ranks number two, with a share of 30%.

Perhaps these figures tell us only what we already know: that Japanese robots are numerous. But by examining current trends, we find something more important: that Japanese robots are getting smarter.

First of all, most quality robot-makers are now switching from 8-bit to 16-bit microcomputer control. Though **Kawasaki** began aggressively marketing its 8-bit spot welding robot this year, it should be selling its 16-bit robots by spring, 1982. A company spokesman explained that 16-bit computers would be restricted to advanced models. The 8-bit machines would still be used in the lower-priced robots.

Japan's top manufacturer of painting robots, **Kobe Steel Ltd.**, is switching to the 16-bit microcomputer for its new arc welding robot. Other manufacturers, including **Hitachi Ltd.**, **Mitsubishi Heavy Industries Ltd.**, **Osaka Transformer Co.**, and **Tokiko Ltd.** also plan to switch from 8-bit to 16-bit machines. I expect that, by the end of 1982, most Japanese quality models of welding, painting, and assembling robots will be equipped with 16-bit microprocessors.

On the research scene, the **Mechanical Engineering Laboratory (MEL)** has developed a new mobile robot. Following a trail of road markings, it moves at a slow 450

meters/hour. At predetermined locations the robot stops and scans the appropriate machinery or meter with its on-board TV camera. Researchers say that MEL's monitoring robot could be used in a radio-polluted nuclear reactor.

MEL receives much of its funds from **Japan's Ministry of International Trade and Industry (MITI)**. From 1982 to 1988, MITI plans to spend about ¥30 billion (\$134 million) sponsoring research leading to "intelligent robots." Among other projects, small and light-weight manipulators, sensory/tactile devices, control devices, and robot programming languages are being actively researched.

Meanwhile, **Fujitsu Fanuc Ltd.** and **Seimens** of West Germany have been developing their own intelligent robot. Model "X" has been in the works since 1980. It is expected to use a CCD sensor and a microcomputer with parallel processing, 16-bit capacity.

Model "X" will be produced and marketed in both Japan and West Germany. Fujitsu Fanuc will look into applying the robot in automotive and machinery components assembly. Seimens will investigate applications in electronics parts assembly. **General Numeric**, a joint Fujitsu/Seimens venture, will handle US marketing from its Chicago office. Initially, the companies planned to introduce Model "X" in 1984. But now Fujitsu Fanuc appears anxious to speed up development and introduce the machine in 1983.

Future Japanese robots will do more than build cars. Researchers are working on a tailoring robot system, which may supercede the human tailor. Sponsored by MITI, the development work will start in 1982, last for seven years, and cost ¥15 billion (\$67 million).

MITI's goal is an almost completely unmanned tailoring system. Robots will help design, then cut, sew, finish, and inspect each garment. This calls for highly responsive, intelligent robots, with human-like visual and tactile senses. Add to this such technology as a computer-aided designing system and an automated pattern making device.

This project will give researchers some interesting technological problems to solve. They will have to develop, for example, an automatic soft handling system and some 3-D vision processing. According to MITI, the development will start with easier-to-make clothing, such as shirts, blouses, and underwear. The more difficult dresses and jackets will follow later.

■

# NEW PRODUCTS



## Assembly Robot Hails from France

Sormel S.A. of Besancon, France, offers its new assembly robot, the Cadratic. Based on an XY indexing table, the Cadratic has up to eight Z axis manipulators. XY movements are controlled by stepping motors, while Z axis actuation is either pneumatic or hydraulic. The Cadratic can simultaneously pick up as many as eight different components and sequentially place them at eight different programmed points. With its onboard Sormel microprocessor—and the appropriate tooling—the robot can be reprogrammed to perform screwing, lubrication, welding, and soldering, as well as insertion operations.

The Cadratic has axis travel lengths of 110mm (X), 250mm (Y), and 60mm (Z). Its maximum speed is 600mm/sec, and it can perform 1800 operations per hour. Its maximum payload is 5kg.

More information is available from Salvatore A.G. D'Agostino, CEERIS International, Inc., 1055 Thomas Jefferson St., N.W., Ste. 414, Washington, DC 20007, 202/342-5400.

**CIRCLE 1**

## Realtime Image Processing

Interactive Video Systems (IVS)

now offers a realtime image processing system. Called the IVS 200, it features up to 768x512x8 bit resolution, realtime (60 Hz) digitization and image storage, high speed interface to a desktop computer (which is included), and a variety of software and peripheral options. Output channels for controlling external devices are also available.

The IVS 200 is compatible with most video sources. It includes image manipulation software, and can be programmed in assembly language or BASIC.

For more information, contact Don E. Yansen, IVS, Inc., 34 South Road, Bedford, MA, 01730, 617 / 275-5569.

**CIRCLE 2**



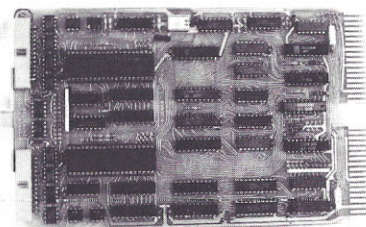
## Support for Speech Synthesis

To support their SP0256 single chip speech synthesizer, General Instrument Microelectronics now has a 16K serial ROM, the SPR016. With this device, a user can add custom programming to the speech synthesizer's onboard ROM.

Constructed on a single monolithic chip, the SPR016 is organized in 8 blocks of 2,048 bits. Two or more chips can be cascaded for larger memory. Operating on a single +5V power supply, the SPR016 can be powered down when the system is inactive. For more information, contact Microelec-

tronics Div., General Instrument Corporation, 600 West John St., Hicksville, NY 11802, 516/753-3606.

**CIRCLE 3**



## Stepper Controller is LSI-11 Compatible

Robotic Synergy is introducing a new line of intelligent multiple-axis stepper motor controllers. The line is called PICMAC (Plug-In, Compatible, Multiple-Axis Controllers). And its first product is the DEC-LSI-11/2, LSI-11/23 Q-bus compatible intelligent stepper motor controller.

This dual-wide card contains two independent four-phase stepper motor controllers. The controllers themselves are microcomputers, able to execute a high-level stepper control language. These stored program devices operate in three modes: immediate command execution, program entry, and stored program execution. In most applications, the stepper motor controller can function without intervention from the LSI-11 host processor, except for program loading. When a stored program is done or the motion is complete, the controller notifies the LSI-11 through interrupts.

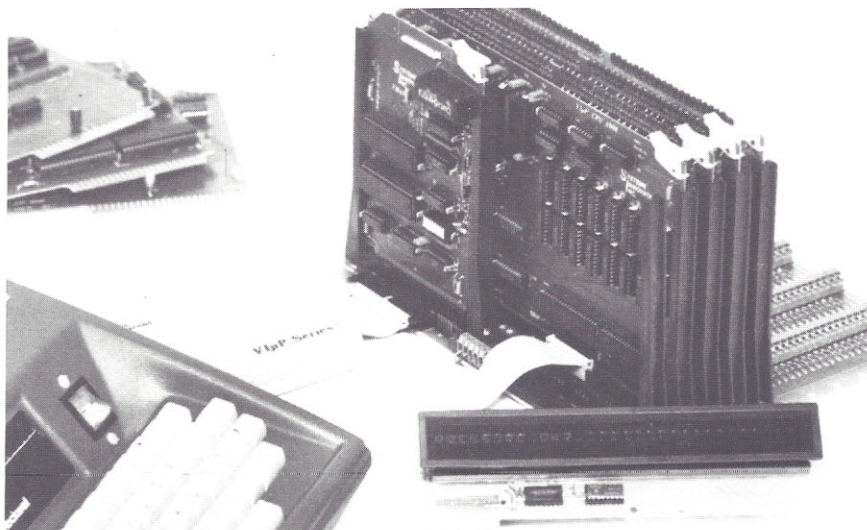
The DEC LSI-11 PICMAC board is available in quantities of 1 to 9 for \$700.00 each. Delivery is stock to six weeks. For more details, contact Robotic Synergy, 13336 So. 1100 East, Ste. 201, Salt Lake City, UT 84105, 801/485-3365.

**CIRCLE 4**

### Stepper Control Kit

Cybernetic Micro Systems now offers a stepper motor control board that uses their CY512 Intelligent positioning stepper motor controller chip. The CY512/KIT consists of a CY512 chip, buffers and LEDs on the chip's output signals, switches on the CY512's inputs, logic for closed loop motor operation, and a 1.5amp/phase unipolar motor drive. All CY512 signals are brought to edge connectors for interfacing to the user's system. The kit also provides a wirewrap area for adding custom circuits.

CY512/KIT comes ready to assemble with complete instructions. It costs \$395 from stock. For more information, contact Cybernetic Micro Systems, 445-203 So. San Antonio Rd., Los Altos, CA 94022, 415/949-0666. **CIRCLE 5**



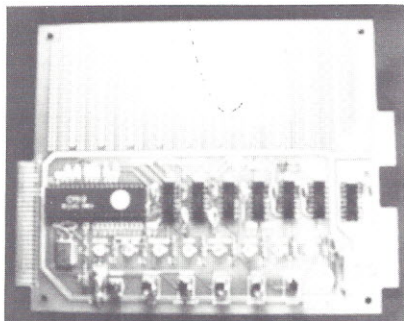
### Microprocessor for Control Applications

Systems Innovations, Inc., is introducing the VI $\mu$ p 7000 (Versatile Industrial Microprocessor). Designed for OEM and small user applications in industrial control, the VI $\mu$ p 7000 can be configured in a number of ways. A configuration can include stepper motor drivers, A/D and D/A converters, a realtime calendar clock, and optically isolated I/O. Using a 6502 microprocessor, the

system bus is equivalent to the KIM 4 and consists of two 44 pin edge card connectors per slot—one for the CPU bus, the other for applications.

The System 7000 is available off the shelf in various configurations. Depending on the configuration, it can cost from \$500 to \$2,000. For more information, contact: Systems Innovations, Inc., P.O. Box 2066, Lowell, MA 01851, 617/459-4449.

**CIRCLE 6**



### Applications Controller

Datronic is now announcing its Applications Controller Series 09. This is a single board system, built around the Motorola 6809 microprocessor on the STD bus. The Applications Controller offers 28-pin sockets, for installing 8-bit wide memory circuits, and a

software-controlled ACIA/MODEM serial port with RS422 or RS232C signalling.

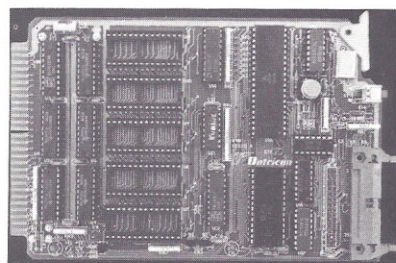
The controller supports memory mapped addressing and I/O page STD bus signalling. Address mapping is accomplished through a programmable ROM that allows users to relocate both onboard and external address assignments.

Datronic has introduced two versions of the controller. One is designated ACS 09-OEM, and the other is call ACS 09-PRO. The ACS 09-PRO is shipped with Datronic's D-Forth language in EPROM and with one 2k $\times$ 8 RAM installed. Other software packages are available.

In quantities of 1 to 24, the ACS 09-OEM lists for \$295. A 2 MHz version, the ACS 09B-OEM, lists for \$340. In the same quantities, the ACS 09-PRO costs \$395. For more information, contact Datronic

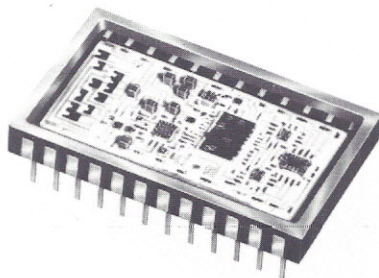
Corporation, 7911 NE 33rd Dr., Portland, OR 97211, 503/284-8277.

**CIRCLE 7**



### Convert Analog to Digital in 900ns

A new high-speed successive approximation A/D converter is now available from ILC Data Device Corporation. Packaged in a hermetically sealed 24 pin DDIP, the



DDC-5101 is a form, fit, and function replacement for the MN 5101 A/D converter.

The DDC-5101 has a conversion time of 900ns and provides nine user-selectable input ranges. Its linearity error is  $\pm \frac{1}{2}$  LSB. No missing codes are guaranteed over the 0 to 70°C or -55 to 85°C operating temperature ranges. The DDC-5101 operates from +15, -15, and +5V power supplies.

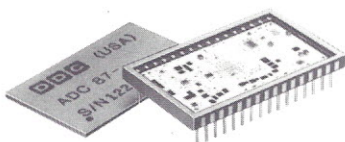
Prices start at \$145 per unit in 100's quantity, with delivery in four weeks. For more information, contact Bill Cullum, ILC Data Device Corporation, 105 Wilbur Place, Bohemia, NY 11716, 515/567-5600.

**CIRCLE 8**

### 12 Bit A/D Converter has Industry Standard Pin Out

Two new 12 bit 10 $\mu$ sec A/D converters, packaged in hermetically sealed 32 TDIPs, are now available from ILC Data Device Corporation. The DDC ADC85 and DDC ADC87 are form, fit, and function replacements for industry standard ADC85 and ACD87 types.

The DDC ADC87 operates over a



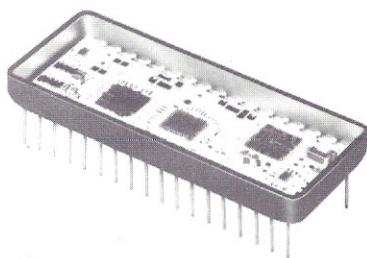
temperature range of -55 to +125°C. Both models feature 1.2 watt power dissipation, 1,600,000 hours of MTBF, and optional MIL-STD-883 screening. Both models have an internal clock and both serial and parallel data outputs. They both operate with  $\pm 15V$  or  $\pm 12V$  power supplies.

In single piece quantities, prices start at \$143. Delivery is stock to 4 weeks. More information is available from Bill Cullum, ILC Data Device Corporation, 105 Wilbur Place, Bohemia, NY 11716, 516/567-5600.

**CIRCLE 9**

### 14 Bit S/D and R/D Tracking Converter

ILC Data Device Corporation is introducing a series of 14 bit synco-to-digital (S/D) and resolver-to-digital (R/D) converters in a single 36 pin DDIP. This is the HSDC-8915



Monobrid Series. Among its features are 10 RPS minimum tracking at 400Hz, 3-state latched outputs for a microprocessor data bus, and low power operation, needing 100mV. Monobrid series converters can also be used as control transformers.

An HSDC Monobrid Converter requires one main +15VDC power supply in addition to the external logic power supply. Internal logic is CMOS, and all logic inputs and outputs are buffered to the external logic level.

The HSDC-8915 series is available

in eight input models, two accuracy grades, and two temperature ranges. Accuracy is 4 minutes  $\pm 0.09$  LSB and  $\pm 2.6$  minutes. Operating temperature ranges are -55 to +125°C and 0 to +700°C.

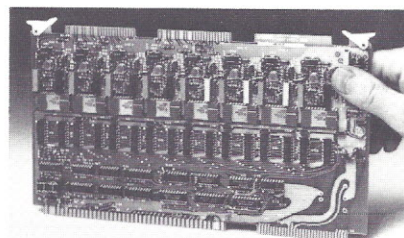
All units are processed to MIL-STD-883, with a computed MTBF of 6,400,000 hours at 25°C. For more information, contact Ken Baker, ILC Data Device Corporation, 105 Wilbur Place, Bohemia, NY 11716, 516/567-5600.

**CIRCLE 10**

### D/A Board Mates to 8 or 16 Bit CPUs

Datel-Intersil now offers the ST-728, a D/A board with 12 bit resolution and—depending on the model—4 or 8 D/A channels. The device is memory mapped, appearing as a block of consecutive memory addresses to the host computer. The base (lowest) address can be located by the user anywhere in memory. Additional ST-728 boards may be added, each at a different base address.

The ST-728 features five selectable output ranges and compatibility with both 8 bit and 16 bit Multibus systems (including the iAPX-86/12 and the SBC-86/12). Input data from the Multibus is double-buffered, permitting simultaneous loading of all 12 bits or serial loading of two successive bytes. The ST-728 draws power from the



Multibus +5V line. It can operate over a temperature range of 0 to +55°C.

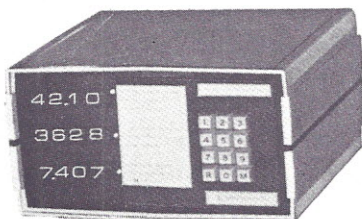
The board comes in three models: ST728A (4 D/A loops), \$695; ST-728B (8 D/A loops), \$875; and ST-728C (4 D/A loops, DC-to-DC converter), \$725. A floppy disk diagnostic and a systems manual are available with each board. Delivery is 2 to 4 weeks. For more information, contact Ted Petit, Datel-Intersil, 11 Cabot Blvd., Mansfield, MA 02048, 617/339-9341. **CIRCLE 11**

### Calculating Process Monitor

A new monitoring and alarm instrument, from Teksym, is designed for OEM production equipment. The calculating process monitor accepts two sensor signals—such as flow, speed, temperature, or force—provides arithmetic processing, then displays current data and generates alarms.

Three 4 digit displays each provide three different display modes. A 16 key pad provides for full numeric and command function inputs. The electrical inputs accept frequencies proportional to sensed parameters. Working in realtime, the internal microcomputer produces current data and keeps running totals for job accounting and inventory control. With an internal rechargeable battery, all scale factors, set points, error bands, and totals are retained during power surges and failures.

For further information, contact



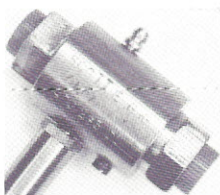
Teksym Corporation, 14504 County Rd. 15, MN 55441, 612/473-1225. **CIRCLE 13**

### Rotating Swivel for Hot Glue

Scott Engineering now offers a 1/4 inch NPT in-line swivel joint that can transfer 375°F hot glue at 1000 psi/10-20 rpm in an automatic gluing machine. The design incorporates a TFE-composition primary balanced seal to minimize torsional friction, a backup seal, and a pressurized grease barrier that isolates the glue from ambient air. A spring-loaded grease cup maintains grease pressure.

More information can be obtained from Scott Engineering Co., 430 N. 9th St., Olean, NY 14760.

**CIRCLE 14**



### 1982 Product Catalog

General Instrument Microelectronics announces its 1982 product selection guide. Included in the new guide will be specifications and brief descriptions of their speech synthesis, video, and control products. The guide also gives a worldwide directory of General Instrument Microelectronics sales offices, distributors, and representatives.

Available at no charge, the catalog can be obtained by writing to Technical Literature Dept., Microelectronics Division, General Instrument Corporation, 600 West John St., Hicksville, NY 11802.

**CIRCLE 16**

## NEW & FREE! 768-Page Handbook Features Small Drive Components



This giant catalog includes over 24,000 off-the-shelf standardized inch and metric belt and chain drives-gears-speed reducers-motors-couplings, universal joints and flexible shafts-countershafts and bearings-special purpose fasteners-vibration mounts-and unique components such as constant force springs.

Learn why thousands of engineers have come to rely on SDP as their single component source for both prototype and production quantities. To get your free copy of SDP Master Catalog 747, just check the Reader Service Number below or contact us today.

**Stock Drive Products** 

Division of Designatronics, Inc.  
55 South Denton Ave.  
New Hyde Park,  
New York 11040  
Tel. (516) 328-3300/328-0200

**CIRCLE 17**

# MEDIA SENSORS

John Markoff, *InfoWorld*, October 12, 1981, **Japan Advances in Speech Recognition Technology.**

Researchers at Japan's Ministry of International Trade and Industry (MITI) have developed systems that can recognize continuous speech at 60 words/minute with high accuracy. This goes beyond the most advanced US systems. IBM, for example, produced a 40 words/minute recognition system, but with low accuracy. Another IBM system could recognize speech from a 1000 word vocabulary with 92% accuracy—but it works 50 times slower than realtime.

The Japanese system, then, is a breakthrough. Although not yet published in Japan, information on the system was reported by N. Richard Miller at the Wescon High Technology Electronics Exhibition and Convention in September. By 1985 or 1986, Miller predicted, the Japanese will be prototyping continuous speech-recognition chips.

High-quality speech recognition may still be as much as a decade away in the US. Still, Miller asserted, discrete phrase-recognition will become increasingly popular in control applications. Speech synthesis will be even more popular, and less expensive. Miller predicts that a speech-synthesis chip will cost only two dollars by 1983.

David Needle, *InfoWorld*, October 12, 1981, **Homes of the Future in Connecticut and Manhattan.**

Two new homes are proving that we can now build the intelligent house.

One of the homes is the

Sun/Tropic House, in Stamford, Connecticut. Though primarily a showcase for solar power and modern design, the house uses an Apple II computer to control its vital functions. Hooked into the home's main antenna, any inside television can act as a control terminal.

The computer monitors a sophisticated stand-alone security system. When the home's security is breached, an outline of the house appears on a terminal, and the location of the breach flashes on and off. At the same time, all window shades go up, and the local police are called.

The home's Apple computer is multitasked to serve as both a control device and a personal computer. Through a telephone modem, the house can even be controlled at a distance.

Meanwhile back in Manhattan, Rose Associates have developed a 53-unit condominium complex, with a computer terminal in every home. Costing from \$128,000 to \$388,000, each unit comes with a Hazeltine Corporation "Esprit" terminal and access to The Source, an information utility. Although compute power in these condominiums is designed mostly for information and entertainment, intelligent home control cannot be far behind.

Margaret Coffey, *Omni*, November 1, 1981, **Robot Evolution.** The next step in human evolution is the robot. So insists Hans Moravec, a researcher at Carnegie-Mellon's Robotics Institute. Within 30 years, he claims, computers will be smarter

and robot bodies better than those of humans.

Humans will transfer their brain patterns to computer programs. But the real fun begins when we start sharing the programs. Then, for example, when we want to build a cabinet, we'll simply borrow the memory of a carpenter.

Ultimately, Moravec says, the concept of the self will blur. As program merges with program, the mind of one human will merge with another. In the end, a single consciousness—representing all intelligent life in the universe—will live in a single cosmic memory bank.

*National Enquirer* June 15, 1981, **One Million People Will Live in Space and Robots Will Have IQs Over 100.** In *The Book of Predictions*, David Wallechinsky, Amy Wallace, and Irving Wallace assemble forecasts of the future from experts in various fields—including robotics. Here are a few robotic predictions that the *Enquirer* excerpted from the book:

- By 1988, most assembly line workers will be replaced by robots.
- By 1989, a computer program will make an original scientific discovery—and be nominated for the Nobel Prize.
- By 1995, you will be able to buy an electric baton and actually conduct a recorded orchestra. The baton will control the music's tempo and volume.
- By 2010, an artificial brain will be built as complex as the human brain, with conscious thoughts and emotions.

The article includes two photographs of radio-controlled "robots" doing household chores. One caption reads: "Robots are already taking out the trash; soon they'll be employed on assembly lines."\*

*The Japan Economic Journal*, October 6, 1981, **Seimans and Fujitsu Fanuc Will Build Unmanned Factory**. By the end of next June, there will be a factory working 24 hours a day, empty of human workers. The Luxemburg plant will be built by Fanuc Mechatronics, a joint venture company of Fujitsu Fanuc Ltd., and Seimans AG of Munich. It will be staffed entirely with industrial robots and automatic parts-carrying equipment.

The factory will produce computer numerical control (CNC) devices at the rate of 50 per month. Its productivity is expected to be 20 times that of a conventional factory.

David Gerrold, *InfoWorld*, September 14, 1981, **Household Robots in Your Future?** "The household robot may be a product whose time has already come and gone."

After all, what chores could it do? It could do the dishes. But that would involve just loading them into the microprocessor-controlled dishwasher. You could use your robot as a secretary. But your home computer already takes care of that. What about a robot to vacuum the floor? A smart vacuum cleaner would do the job more cheaply.

Do you see the problem? By the time we can build a household robot, special purpose devices may be doing all the work we could give it.

Still, the idea of having a household robot is too fascinating to pass by. If only we had a job for it. Fortunately, there is at least one boring, slow, time-consuming task a robot can easily handle. For this job alone, the author would purchase a robot—if only to watch the expressions on the neighbor's faces.

The household robot will do the windows.

*\*Our Editorial Director, Alan Thompson, made predictions for the book. However, none of them were excerpted by The Enquirer.*

**Media Sensors** are brief summaries of robotics-related items that have appeared in the mass media. An attempt is made to paraphrase the content of the original item without altering the tone. The views expressed in these items are not necessarily those of *Robotics Age*. If you have an item you would like to contribute, send it along with a complete identification of its source, to:

Media Sensors  
Robotics Age  
P. O. Box 725  
La Canada, CA 91011

## Calendar

(cont. from pg.53)

obtained from David Smith, 4505 Kennedy Blvd., North Bergen, NJ, 07047, 201/865-4890.

*Previously announced:*

*Refer to the original Robotics Age Calendar for details.*

**Robots VI**, March 1-4, 1982, Detroit, MI, Announced in Vol. 3, No.4.

# CLASSIFIED

**6502 PROGRAMMABLE CONTROLLER**—The TimeStack software/hardware expansion converts a KIM-1 into a general purpose programmable controller. Up to 30 events possible with auto-repeat. Adaptable to other 6502 systems such as OSIC1P, AIM-65, and VIC-20. Specify system. Software manual \$15, Hardware manual \$5. Send 35¢ stamp for more info. Hunter Technical Services, PO Box 359, Elm Grove, WI 53122.

**SURPLUS**—Parts for robot: Pneumatic and hydraulic cylinders, motors and fittings. Electrical DC torque motors, servo amplifiers, tachometers, stepper motors, sonic sensors, rate gyros, sprockets and chain, pulleys and belts, plus many more.

Some parts are arranged as kits. We supply diagrams and design ideas. You supply the common materials found in your local hardware store. The majority of our parts are unused like new. Send for our catalogue: PM Custom Electronics; PO Box 322; Needham, MA 02192.



Our classified advertising section is for readers wishing to buy, sell or trade hardware of software. The price is 30¢ per word. The first word is set in all caps. Minimum 20 words. Send copy with check to ROBOTICS AGE CLASSIFIED, PO Box 725, La Canada, CA 91011.

# ORGANIZATIONS

## First LISP Machines Delivered

LISP has long been the programming language favored by Artificial Intelligence (AI) researchers. Lately, though, the MIT-born language has begun branching out to the commercial world. Two companies—Symbolics and Lisp Machines Inc. (LMI)—have each developed computer systems designed specifically to run LISP.

Both companies have a non-exclusive license from MIT. But the competitors are not offering identical goods. Symbolics has modified MIT's product, while LMI is now marketing MIT hardware with no major changes. LMI seems to have reached the market first. Its first deliveries of the Series III LISP machines were recently accepted by Texas Instruments and CDC.



LMI's Series III LISP Machine.

Meanwhile, LMI has signed an agreement with Western Digital to jointly develop a new series of LISP machines. They plan to base the first system on the computer Western Digital is now developing, also under license from MIT. A company

## Calendar

LTI Robotics Systems will be holding three one-day **Workshops on Advanced Robotics for Assembly**, on December 8, 9, and 10. The workshops will include demonstrations of hardware and lectures on robotic work cells. They will also provide users with hands-on experience. For complete details, contact Jerry W. Saveriano, LTI Robotics Systems, 2701 Toledo St., Suite 701, Torrance, CA, 90503, 213/328-4051.

**Meetings of the International Robotics Foundation (IRF)** will be held the second Monday of each month at 7:00 PM. The meetings will take place at Cal State Long Beach, Data Processing Department, Room 180D. For further information, contact IRF, P.O. Box 3227, Seal Beach, CA, 90740.

**Monthly Robot Club Meetings** will be held on the last Thursday of each month, from 7:00 to 9:00 PM. The meetings will take place at Hepburn Hall, Jersey City State College, 2039 Kennedy Boulevard, Jersey City, NJ. Future meeting times and places may change. But complete details can always be

*(continued pg.52)*

spokesman said that the new system should be in volume production by 1982. As the number of intelligent systems increase, so, we expect, will the number of LISP machine users.

## ROBOT RESEARCH CAREERS

The growth field of Robotics has led to many varied opportunities for faculty and laboratory staff. Opportunities exist for state-supported positions and positions supported through industrial funds and federal grants. People with skills and interests in one or more of the following are needed: COMPUTERS, VISION, SENSORS, ROBOT CONTROL, ALGORITHMS, SOFTWARE ENGINEERING, TECHNICAL WRITING, RESEARCH COORDINATION, ECONOMICS, KINEMATICS, COMPUTER AIDED DESIGN, ELECTRONICS, MACHINE INTERFACE, HANDS, MACHINE INTELLIGENCE, INSPECTION, COMPUTER LANGUAGES. Full and Part Time Employment is also available for people with varied educational background currently pursuing a BS, MS, or PhD in EE, ME, or COMPUTER SCIENCE. Send resume, stating background, performance, and references to: Professor John R. Birk, Director, Robot Research Group.

College of Engineering  
**UNIVERSITY OF RHODE ISLAND**  
Kingston, Rhode Island 02881

An Affirmative Action/Equal Opportunity Employment m/f

**CIRCLE 31**

## CALL FOR ARTICLES

*We Know You're Out There—*

We can hear your curtains closing under microprocessor control. We can see your room lights blink off and on at the command of your voice. We know that some of you—our loyal readers—are using your home computers to control devices in your houses and cars.

We know you're out there, and we'd like to hear more from you. We need articles on using controllers in the home, on interfacing household devices to home computers, and on using microprocessors to monitor or improve the performance of the family car.

Intelligent control systems are at the forefront of home robotics. If you are working in these areas, our editors—and our readers—want to read about your work. Our rates for accepted manuscripts are among the highest in the field. We pay up to \$50 per magazine page.

Send your manuscripts or enquiries to: Editor, **Robotics Age**, PO Box 725, La Canada, California 91011-0801.

# DESKTOP 68000 + TELESOFT<sup>1</sup> ADA<sup>4</sup> + PASCAL = **POWER**

## THE IN/7000D BRINGS OUT OF THIS WORLD A/I PERFORMANCE WITHIN REACH

### 68000 BASED IN/7000D

Dramatically increased processing speed and flexibility:

- 8 MHz 68000 CPU
- 32-Bit internal arithmetic registers
- 24-Bit address register
- Powerful assembly language instructions support modular programming
- Designed for expansion to 32 bit word size
- 8 Levels of interrupt priority
- Vectored interrupts and DMA fully supported
- Outbenchmarks the IBM 370/145<sup>2\*</sup>
- High speed string processing

### Vastly increased memory:

- 68000 directly addresses 16 MB of memory
- Sorts can be done in core rather than with disk I/O

### PASCAL AND TELESOFT<sup>1</sup> ADA<sup>4</sup>

- Fast program development
- Self documenting
- Supports structured programming
- Easy to update
- Easy to maintain
- Transportable from computer to computer
- Powerful logical constructs greatly simplify programming

For example: Modular procedures and functions  
Strong data types  
Case structures

- Local and global variables
- Recursive problem solving
- Block insert — a block of statements may be inserted anywhere one statement can exist
- Built in Boolean functions
- Library capability
- Program segmentability
- Procedure linking
- TeleSoft<sup>1</sup> Pascal and TeleSoft<sup>1</sup> Ada<sup>4</sup> translate to 68000 native code
- Built-in powerful string-handling features

### TELESOFT<sup>1</sup> ADA<sup>4</sup>

- Designed to fulfill all DoD specifications
- INTRINSIC functions include:
  - \* Multi-tasking and multi-programming
  - \* Independent compilation of program units (called Packages)
- Fully implemented syntax checker which now parses the entire Ada<sup>4</sup> language

COMPATIBILITY WITH DEC Q-BUS<sup>3</sup>  
AND STANDARD DEC<sup>3</sup> PERIPHERALS

### Saves development time.

- High speed sensory data processing
- High speed string processing power
- Fast coordinate transformations
- Easy implementation of in-memory AI algorithms, predicate calculus and trajectory computations
- Design and test algorithms quicker and easier
- Both Pascal and 68000 support features that make debugging far more efficient
- Plenty of memory, no need to use extra time for "programming tricks," previously needed with limited memory
- Mixed mode listing (Pascal source statements followed by 68000 statements)

UP TO 4 MByte OF RAM

<sup>1</sup>"Kilobaud Microcomputing" October, 1980

<sup>2</sup>A trademark of Renaissance Telesoftware Inc.

<sup>3</sup>A trademark of International Business Machines

<sup>4</sup>A trademark of Digital Equipment Corporation

<sup>5</sup>A trademark of Department of Defense

Call us now for more information, or fill in the coupon (301) 840-0700

### I AM INTERESTED! Send more information now.

Name \_\_\_\_\_ Title \_\_\_\_\_

Company \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone \_\_\_\_\_ Call me IMMEDIATELY ☐

My need is: ☐ 1-3 mo. ☐ 3-6 mo. ☐ Future project ☐ File

☐ I am interested in a dealership.

### INTELLIMAC/Intelligent Machines

51 Monroe St., 18th Floor • Rockville, Md. 20850  
Ph. (301) 840-0700 • TWX: 710-828-9786 • Cable: EAIL  
ROCKVILLE MD

**INTELLIMAC**  
*Intelligent Machines*

51 Monroe St., 18th Floor  
Rockville, Md. 20850  
(301) 840-0700  
TWX: 710-828-9786  
CABLE: EAIL ROCKVILLE MD

Dealerships are available

© INTELLIMAC 1981

SATURN PHOTOGRAPH COURTESY OF NASA